

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2004年4月15日 (15.04.2004)

PCT

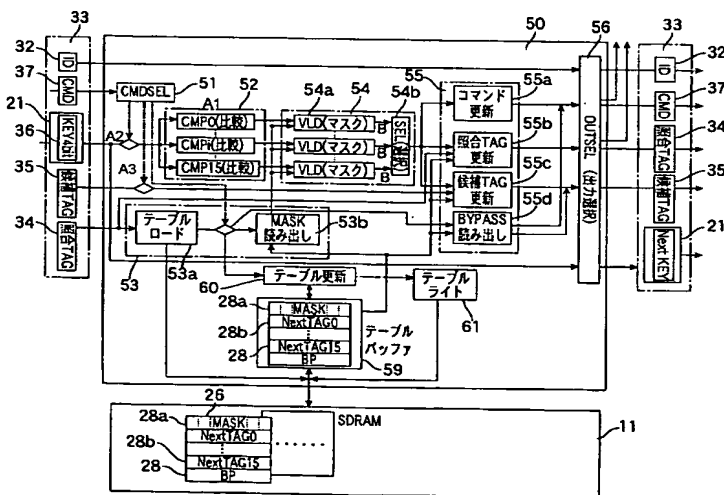
(10) 国際公開番号
WO 2004/032435 A1

- (51) 国際特許分類: H04L 12/56, G06F 17/30 (72) 発明者; および
(21) 国際出願番号: PCT/JP2003/012700 (75) 発明者/出願人 (米国についてのみ): 佐藤 哲朗
(22) 国際出願日: 2003年10月3日 (03.10.2003) (SATO,Tetsuro) [JP/JP]; 〒206-0034 東京都多摩市鶴
(25) 国際出願の言語: 日本語 牧 6丁目16番地4-502 Tokyo (JP). 河口 文法
(26) 国際公開の言語: 日本語 (KAWAGUCHI,Fuminori) [JP/JP]; 〒177-0044 東京都
(30) 優先権データ: 特願2002-291708 2002年10月3日 (03.10.2002) JP 練馬区上石神井2丁目6番12号 Tokyo (JP).
(71) 出願人 (米国を除く全ての指定国について): 株式会社
社インフォーエス (IN4S INC.) [JP/JP]; 〒103-0007 東
京都中央区日本橋浜町三丁目16番7号 Tokyo (JP).
(81) 指定国 (国内): AE, AG, AL, AM, AT, AU, AZ, BA, BB,
BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK,
DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR,
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,

[続葉有]

(54) Title: BIT STRING CHECK METHOD AND DEVICE

(54) 発明の名称: ビットストリングの照合方法および装置



35...CANDIDATE TAG
34...CHECK TAG
A1...CMPO (COMPARISON)
A2...CMP1 (COMPARISON)
A3...CMP15 (COMPARISON)
B...VLD (MASK)
54b...SEL (SELECTION)
55a...COMMAND UPDATE
55b...CHECK TAG UPDATE
55c...CANDIDATE TAG UPDATE
55d...BYPASS READ OUT
56...OUTSEL (OUTPUT SELECTION)
53a...TABLE LOAD
53b...MASK READ OUT
60...TABLE UPDATE
61...TABLE WRITE
59...TABLE BUFFER

(57) Abstract: In a bit string check method, a bit string to be searched is divided into a plurality of partial object bit strings, which are compared to the registered bit patterns at multiple stages. At a current stage which is one of the multiple check stages, comparison is made with all the values which can acquire the partial object bit string. By using the comparison result and the pattern table where a partial registration bit pattern is registered, it is possible to obtain a judgment result indicating presence/absence of a partial registration bit pattern which coincides with at least the partial object bit string. According to the judgment result, check continuation information including the address of the pattern table of the stage subsequent to the current stage is output.

[続葉有]



NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

- (84) 指定国 (広域): ARIPO 特許 (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア特許 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

規則4.17に規定する申立て:

- すべての指定国のための先の出願に基づく優先権を主張する出願人の資格に関する申立て(規則4.17(iii))

添付公開書類:

- 国際調査報告書

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(57) 要約:

本発明のビットストリングの照合方法においては、検索対象のビットストリングを複数の部分対象ビットストリングに分けて、登録されたビットパターンと多段階で照合する。多段階の一つの照合段階である現段階では、部分対象ビットストリングを取り得る全ての値と照合し、前記照合の結果と、部分登録ビットパターンが登録されたパターンテーブルとにより、少なくとも部分対象ビットストリングに一致する部分登録ビットパターンの有無を示す判定結果を得て、前記判定結果により、現段階の次の段階のパターンテーブルのアドレスを含む照合継続情報を出力する。

明 細 書

ビットストリングの照合方法および装置

5 技術分野

本発明は、IPアドレスなどのビットストリングを検索あるいは照合に適した装置および方法に関するものである。

背景技術

- 10 インターネットなどのコンピュータネットワークを介してデータを送受信する際には種々のプロトコルが用いられる。その1つは、インターネットのネットワーク層におけるプロトコルであり、インターネットプロトコル（IP）と呼ばれている。このIPは、情報をパケット化して所定の目的地まで移動させるが、そのために、情報を格納したペイロードにIPヘッダをつけたパケットを、情報の送受信のために用いる。各々のパケットは、IPヘッダの情報に基づいて、所定のルートに配送され、受信先では、IPヘッダの情報に基づいてペイロードの情報が送信元の状態となるように組み立てられる。

- IPヘッダの情報は、ルータおよびファイアウォールなどのネットワークを管理する装置あるいはソフトウェア（プログラムあるいはプログラム製品）で参照され、ルーティング、パケットフィルタリング、SPD（セキュリティ・ポリシーデータベース）、ポリシールーティング、QoSなどの様々な目的のために照合される。このように、IPヘッダの情報は、パケットの送受信あるいは転送については常に参照され、利用される。

- 今後、モバイルIPが実現されると、ルーティングテーブルは動的に変動する可能性はある。また、現状のIPv4（バージョン4）では、受信先アドレス（SIP）および送信元アドレス（DIP）は32ビットであるのに対し、IPv6（バージョン6）では128ビットに拡張される。したがって、IPアドレスの検索における処理速度をさらに向上させることが、インターネットなどのコンピュータネットワークを用いてデータ交換をする上で、今後、さらに重要な技

術となる。

たとえば、特開平 1 1－8 8 4 2 7 号公報には、ソートされた複数のエン트리からなるルーティングテーブルに対して検索 I P アドレスを大小比較しながら一致する I P アドレスを検索する技術が開示されている。この方法によりルーティ
5 ングあるいはその他の目的のために 1 2 8 ビットの I P アドレスを複数含むビットストリングを検索しようとする膨大な時間が必要になり、少なくとも、今後の情報伝達経路に用いるには適していない。

I P アドレスに限らず、検索対象となるビットストリング（ビットストリームあるいはビットパターン）の情報量は、今後、さらに増大する傾向にあり、光通
10 信が本格化する場合は、G（ギガ）ビットクラスのビットストリングの検索機能が要求される。膨大なデータ量のビットストリングの検索方法は幾つか提案されている。

まず、ソフトウェア方式は、検索機能は十分だが、速度的には不十分である。ハードウェアを用いた方式では、C A M（Content-Addressable Memory）方式が
15 ある。1 個の C A M を使った繰り返し検索により最長一致検索を行ったり、複数個の C A M を並列に使っての最長一致検索を行うことは現実的である。しかしながら、パケットフィルタリングやステートフルインスペクションに要求される多次元範囲、たとえば、ソース I P ・デスティネーション I P ・ソースポート・デ
20 スティネーションポート・プロトコルを含む範囲の一致検索を実現するのは非常に困難である。

また、最長一致検索の対象として I P ルーティングを考えた場合、C I D R（Classless Inter-Domain Routing）形式での可変プレフィックス長への対応は、繰り返し型の繰り返し回数を増やすか、多数個 C A M 並列型でおこなうことになるであらう。前者であればスループットやレイテンシ面での性能が不足すること
25 になる。後者であれば経路エントリの分布ばらつきにより、C A M エントリ利用率が悪くなるという問題点がある。

登録エントリ数の増加に対しては複数 C A M チップのカスケード接続でしか対応できない。被検索キーのビット長の増加に対しては、チップ自体のスペック変更でしか対応できず、全エントリの C A M メモリビット増が必要である。多チッ

ブ構成で対応しようとする、CAMは多数のエントリが並列に照合されるため、複数のチップ間で連動した照合をおこなうには各エントリの照合状態をチップ間で継承しなければならない。したがって、CAMに求められる大量のエントリ数を考えると現実的ではない。

- 5 部分的にCAMを利用する方法もある。たとえば、CAMにより検索対象のエントリを絞り込み、その後、パトリシアツリー等をソフトウェアにより検索するという方式がある。この方式では、ツリー検索にソフトウェアを用いるため、完全ソフトウェア方式と同様の問題がある。また、ツリー検索部をハードウェア化した場合でも、例えばツリー構造がパトリシアであれば、エントリの登録および
- 10 /または削除に複雑なツリー再構成が伴い、上位アプリケーションソフトウェアに過度なテーブル管理コストを強いることになる。

- マスキングが可能で、階層型の検索を可能としたターナリーCAM (TCAM) 方式がある。この場合、ハードウェアそのものが提供する検索機能は、完全一致 (CAM) と最長一致である。TCAMは原理的にはエントリのDC (ドン
- 15 トケア) ビットをも一致と見なす完全一致であり、照合部からは複数の一致エントリが出力される。そのため、照合部の後段に、プレフィックス長あるいはエントリNo等による何らかのプライオリティエンコードがないと、「複数ヒットから最長一致」の絞り込みをおこなえない。

- 従って、TCAMにおいてはエントリの登録および/または削除についての管理が煩雑となる。特に、エントリNoがプライオリティとなる方式のTCAMの場合、エントリの追加にともない、既存エントリの再配置を上位アプリケーションソフトウェア側が自発的に行わねばならない。この既存エントリの再配置はプレフィックス長をキーとしたソートであり、複数回のエントリ位置交換が要求される。プレフィックス長だけが問題なので通常のソートよりは低コストであるが、
- 20 ソートそのものは排除できない。すなわち、TCAMで範囲一致は実現できるが、それはエントリ管理をおこなう上位アプリケーションソフトウェアによるものである。したがって、範囲表現を複数の最長一致表現へと分割し、最長一致へ還元して実現することになる。そのため、範囲一致のルール管理 (削除・登録) が煩雑となる。つまり、TCAMにおける範囲一致は、静的なエントリからなる検索

については実用性があつたとしても、動的に変化するエントリからなる検索については、上位アプリケーションソフトウェアへの負荷がかなり高いといえる。あるいは、これら複雑なテーブル管理をおこなう専用のコプロセッサをTCAM自体とは別途用意することになる。

- 5 以上のことから、範囲一致は無論、最長一致であっても、登録エントリがダイナミックに追加削除されていく性質のある検索にはTCAMは向かないといえる。登録エントリ数の増加に対しては、複数のTCAMチップのカスケード接続でしか対応できないことはCAMと同様である。被検索キーのビット長の増加に対しても、チップ自体のスペック変更でしか対応できず、全エントリのTCAMメモリビットを増加する必要がある。多チップ構成で対応しようとするれば、TCAM
- 10 は多エントリが並列に照合されるため、複数チップ間で連動した照合をおこなうには各エントリの照合状態をチップ間で継承しなければならない。したがって、TCAMに求められる大量エントリ数を考えると現実的ではない。

- そこで、本発明の目的の1つは、範囲一致を始め、完全一致および最長一致と
- 15 いった検索機能を提供可能であり、パケットフィルタリングやステートフルインスペクションにおいても、上位アプリケーションソフトウェアの負担が軽いビットストリングの照合方法および照合装置を提供するである。エントリの追加・削除にともなうエントリ移動により発生する、上位アプリケーションによるテーブル管理のオーバーヘッドを縮小または排除できるビットストリングの照合方法および照合装置を提供することも本発明の目的の1つである。
- 20

- さらに、メモリとして、SDRAM等の通常のDRAM（あるいはSRAM）を使用可能とし、十分な検索速度を提供しながら、CAM系の専用メモリよりもはるかに廉価で大容量を実現できる照合方法および装置を提供することも本発明の目的の1つである。そのような照合装置は、収容可能なエントリ数はSDRAM容量にのみ規定されるため、スケーラブル性が高い。また、メモリ管理が柔軟に行えるため、複数種の検索を単一のハードウェアセットで柔軟に行えるものになる。
- 25

 また、検索対象となる被検索ビットストリング（キー）のビット長の増加に対しても、ハードウェア自体のアルゴリズム的な変更は不要で、ユニットを追加し

たり、一連の処理を繰り返すことにより対応可能な照合方法および照合装置を提供することも本発明の目的の1つである。

発明の開示

- 5 本発明のビットストリングの照合方法においては、検索対象のビットストリング（被検索ビットストリング）を複数ビットの部分対象ビットストリング（部分被検索対象ビットストリング）に分けて、予め登録された複数の登録ビットパターンと多段階で照合するプロセスを有する方法を採用する。多段階に含まれる1つの照合段階である現段階は、検索対象のビットストリングから、現段階の部分対象ビットストリングを選択し、その現段階の部分対象ビットストリングが取り得る全ての値と照合する全照合工程と、複数の登録ビットパターンの部分登録ビットパターンを示すためのパターンテーブルであって、現段階に先行する段階から得られる照合継続情報により決まる現段階のパターンテーブルを、全照合工程と前後して、または並列に、メモリよりロードするパターンロード工程とを有する。この現段階のパターンテーブルは、現段階の部分対象ビットストリングに対して照合対象となる部分登録ビットパターンを示す。
- 10
- 15

- 現段階は、さらに、全照合工程の結果と現段階のパターンテーブルとにより、現段階の部分対象ビットストリングに一致する部分登録ビットパターンの有無を少なくとも示す照合結果を得る判定工程と、この照合結果により、現段階のパターンテーブルに対応するアドレステーブルから、現段階の次の段階のパターンテーブルのアドレスを含む照合継続情報を出力する出力工程とを有する。アドレステーブルには、現段階のパターンテーブルに示された部分登録ビットパターンのそれぞれに続く次の段階のパターンテーブルのアドレスが示されている。このため、アドレステーブルの中から、現段階において一致する部分登録ビットパターンに続く、次の段階で照合する部分登録ビットパターンを示す次の段階のパターンテーブルのアドレスを含む照合継続情報を出力することができる。
- 20
- 25

この照合方法を実現する照合装置として、本発明においては、検索対象のビットストリングを複数ビットの部分対象ビットストリングに分けて、予め登録された複数の登録ビットパターンと多段階で照合するための照合装置であって、多段

階の少なくとも1つの照合する段階を実行する照合装置を提供する。この照合装置は、検索対象のビットストリングから、多段階の1つの現段階の部分対象ビットストリングを選択し、その部分対象ビットストリングが取り得る全ての値と照合する全照合手段と、先行する段階から得られる照合継続情報により決まる現段階のパターンテーブルをメモリよりロードするパターンロード手段と、全照合手段の結果と現段階のパターンテーブルとにより、部分対象ビットストリングに一致する部分登録ビットパターンの有無を少なくとも示す照合結果を出力する判定手段と、照合結果により、現段階のパターンテーブルに対応して記録されたアドレステーブルから、次の段階のパターンテーブルのアドレスを含む照合継続情報を出力する出力手段とを有する。

これらの照合方法および照合装置は、現段階で一致する部分登録ビットパターンに続く次の段階のパターンテーブルのアドレスを含む照合継続情報を出力することにより完全一致および最長一致の検索機能を提供する。さらに、判定工程および判定手段において、部分対象ビットストリングに値が最も近い最大または最小の部分登録ビットパターンの有無を含む照合結果が得られると、出力工程および出力手段では、一致する部分登録ビットパターンがないときに、アドレステーブルから、最大または最小の現段階の部分登録ビットパターンに続く次の段階のパターンテーブルのアドレスを含む照合継続情報を出力できる。これにより、検索機能として範囲一致を含めた照合方法および照合装置を提供できる。

ここで、部分対象ビットストリングに値が最も近い最大の部分登録ビットパターンとは、部分対象ビットストリングより値が小さくて最大の部分登録ビットパターンを意味する。また、部分対象ビットストリングに値が最も近い最小の部分登録ビットパターンとは、部分対象ビットストリングより値が大きくて最小の部分登録ビットパターンを意味する。最大の部分登録ビットパターンに続くパターンテーブルのアドレスが選択されることにより、以下で説明する右倒れの範囲検索が可能となる。また、最小の部分登録ビットパターンに続くパターンテーブルのアドレスが選択されることにより、以下で説明する左倒れの範囲検索が可能となる。したがって、右倒れ型の検索においては、以下で説明する最大の側が選択され、左倒れ型の検索においては、以下で説明する最小の側が選択される。

また、本発明の照合方法および照合装置においては、多段階の照合の1つの段階（現段階）において、全照合工程および全照合手段により、部分対象ビットストリングが取り得る全ての値と照合する。その結果と、現段階の部分対象ビットストリングと照合する対象となる、現段階の部分登録ビットパターン（以降では
5 エントリとも称する）を示す現段階のパターンテーブルとから完全一致または範囲一致を導く。

パターンテーブルは、部分エントリのビット列そのものである必要はなく、むしろ、部分エントリの値を直接またはビットパターンあるいはフラグなどにより
10 部分エントリと対応するような形式で示すデータであることがデータ量の点から望ましい。本発明においては、部分対象ビットストリングが取り得る全ての値と照合する全照合工程あるいは全照合手段を備えている。したがって、全照合の際に照合の対象となる値の範囲において、パターンテーブルは、部分登録ビットパターンの有効無効を示すビットフラグからなるマスクデータで与えることができる。
15 すなわち、全照合の際に、予め部分対象ビットストリングが取り得る値の範囲が決まっているので、そのようなマスクデータでメモリに記憶し、またメモリからロードすることが可能である。したがって、パターンロード工程および手段においては、マスクデータをメモリからロードする。

また、この照合方法および装置においては、部分対象ビットストリングが取り
20 得る全ての値と照合するので、パターンテーブルの量あるいは領域としては、部分対象ビットストリングが取り得る全ての値をカバーできる量あるいは領域が予め確保される。したがって、エントリを追加する際は、基本的にはマスクパターンを更新するだけで良く、エントリの追加削除にともなうエントリ移動は発生しない。このため、上位アプリケーションによるテーブル管理のオーバーヘッドを縮
25 小または排除できる。したがって、パケットフィルタリングやステートフルインスペクションにおいても上位アプリケーションソフトウェアの負担が軽いビットストリングの照合方法および照合装置を提供できる。

特に、マスクデータの場合は、1ビットで1エントリを表現することも可能となるので、パターンテーブルの記憶容量は大幅に削減され、また、パターンテー

ブルをロードする時間も短縮される。さらに、更新する際も、エントリの追加削除の処理に書き換えるビット量は大幅に少なくなり、処理は簡略化され、処理時間も短縮される。したがって、パターンテーブルをマスクデータ化することは有効である。

5

部分対象ビットストリングが取り得る全ての値と照合する処理（全照合工程および手段）は、比較器やルックアップテーブルを用いてハードウェアで高速に実行可能であり、多段階に分割することによりハードウェアの規模も大きくなり
10 長が長くなればハードウェアは大きくなり、ビット長が短ければ照合段数が増加する。ハードウェアの適切な規模は、ハードウェアの目的、経済的な価値、実現可能な配線ルールなどに依存して変わる。現状では、部分対象ビットストリングのビット長としては4から5ビット程度が好ましいと考えられる。さらに、検索対象となる被検索ビットストリング（以降ではキーとも称する）のビット長増に
15 対してもハードウェア自体のアルゴリズム的な変更は必要なく、ユニットを追加したり、一連の処理を繰り返すことで対応可能となる。

本発明の照合方法および照合装置は、簡易な構成で高速化できるという大きな効果を備えている。まず、全照合工程とパターンロード工程とを並列に行うこと
20 が可能であり、多段階に分割することに伴い発生するメモリからパターンテーブルをロードするオーバーヘッドの時間も大幅に削減できる。したがって、特殊な構造で高速なメモリを使わなくても、十分に速い検索速度を発揮でき、廉価で大容量の検索を高速に実行する照合方法および装置を提供できる。そのような照合装置は、収容可能なエントリ数はSDRAM容量にのみ規定されるため、スケーラ
25 ブル性が高く、また、メモリ管理が柔軟に行える。したがって、複数種の検索を単一のハードウェアセットで柔軟におこなえるものになる。

メモリに対してデータを入出力するキャッシュ装置を設けることにより、パターンテーブルをロードする時間のオーバーヘッドをさらに削減できる。キャッシュメモリの効率を向上し、ミスヒット率を低減するには、パターンテーブル、

特にマスクデータ、およびアドレステーブルを含み、メモリ上の記憶単位である照合テーブルのサイズとキャッシュラインのサイズが同一であることが好ましい。

さて、範囲一致検索においては、現段階では、一致する部分登録ビットパターンがあり、さらに、最大または最小の部分登録ビットパターンがあるときには、最大または最小の部分登録ビットパターンに続く次の段階のパターンテーブル（範囲検索のパターンテーブル）のアドレスを候補アドレスとして記憶あるいは出力しておくことが望ましい。照合継続のときの現段階（候補アドレスを記憶した照合段階に対しては次の照合段階）において、一致する部分登録ビットパターンがなく、最大または最小の部分登録ビットパターンもないときには、最大または最小の部分登録ビットパターンが発生する段階まで遡るバックトラックが発生する。そのときに、候補アドレスが記憶されていれば、照合継続の現段階の出力工程において、先行する段階における候補アドレスを含む照合継続情報を出力することにより、バックトラックにより照合を継続する段階まで最短の処理時間で到達できる。

範囲一致検索において、最大または最小の部分登録ビットパターンに続く現段階のパターンテーブル（範囲検索のパターンテーブル）のアドレスを含む照合継続情報が与えられたときは、現段階のパターンテーブルに示された最大または最小の部分登録ビットパターンに続く次の段階のパターンテーブルのアドレスを辿ることにより、検索対象のビットストリングよりも値が小さくて最大の登録ビットパターン（最大の登録ビットパターン）、または、検索対象のビットストリングよりも値が大きくて最小の登録ビットパターン（最小の登録ビットパターン）に到達することができる。したがって、その状態、すなわち、範囲検索のパターンテーブルが得られた現段階では、全照合工程および判定工程をバイパスすることができる。そして、出力工程および出力手段により、範囲検索のパターンテーブルを現段階のパターンテーブルとし、それに対応するアドレステーブルから、そのパターンテーブルに示された最大または最小（パターンテーブルの中での最大または最小）の部分登録ビットパターンに続く次の段階のパターンテーブルのアドレスを含む照合継続情報を出力する処理を繰り返せば良い。

このように段階毎のパターンテーブルの最大または最小を辿りながら最大または最小の登録ビットパターンにたどり着く処理は、エントリの登録、削除がない限りパターンテーブルが決まれば、たどり着く登録ビットパターンは同一になる。したがって、パターンテーブルと共に、そのパターンテーブルに示された最大また

5 最小の部分登録ビットパターンに対応して決まる最大または最小の登録ビットパターンを示すバイパスデータをメモリに記憶しておくことが望ましい。最大または最小の部分登録ビットパターンに続く範囲検索のパターンテーブルのアドレスを含む照合継続情報が与えられたときは、全照合工程、パターンロード工程および判定工程をバイパスし、出力工程および出力手段において、範囲検索のパ

10 ターンテーブルに対応するバイパスデータを含む最終照合情報を出力して検索対象のビットストリングの照合を終了することができる。したがって、中間の段階のパターンテーブルを辿る処理を省くことができる。

特に、上述したバックトラックが発生したときでも、その後の処理は候補アドレスに示されたパターンテーブルに対応するバイパスデータを出力するという処理で終了し、ほとんどバックトラックによるペナルティーは発生しない。

15

本発明の照合方法および照合装置は、さらに、検索対象となるビットストリングのビット長増に対してもハードウェア自体のアルゴリズム的な変更は必要なく、照合ユニット、すなわち、多段階の1あるいは複数段階を実行する照合装置を追加したり、上述した現段階の一連の工程を繰り返すことで対応できる。したがって、本発明において、複数の照合段階を直列および／または並列に行う分類過程を有する分類方法は、検索目的、検索対象のビットストリングの種類、および登録ビットパターンの種類などの条件に応じて極めてフレキシブルに設計できる。同様に、複数の照合装置を直列および／または並列に組み合わせた分類装置も目的に応じて極めてフレキシブルに設計できる。

20

25

これらの分類過程あるいは分類装置においては、複数の照合段階あるいは照合装置をパイプライン方式で接続することにより、検索時間の実質的な短縮を実現でき、この点でも、検索処理の高速化を図ることが容易である。すなわち、複数の照合段階または照合装置の間では、照合継続情報を含むデータをパケット化し

て伝達することにより、データフロー型の処理で検索処理を実行できる。

さらに、本発明の照合方法および照合装置においては、登録ビットパターンを追加あるいは削除する際に、エントリの移動は発生せず、単に、パターンテーブルおよびアドレステーブルを更新することにより、登録ビットパターンを追加または削除できる。したがって、そのような更新工程を実行する更新手段を設けておくことにより、更新するためのデータをパケットとして上流から下流に流して、エントリの追加削除を行うことができる。したがって、パイプライン方式で実行される、それぞれの照合段階の照合テーブルを更新する処理（更新過程）もパイプライン方式により照合段階と並列に行うことができる。すなわち、本発明においては、パイプライン方式で実行される、それぞれの照合段階の照合テーブル（パターンテーブルおよびアドレステーブルを含む）を更新することにより、登録ビットパターンを追加または削除する更新過程を有し、この更新過程は、照合段階と並列にパイプライン方式で実行できる。

照合テーブルが、パターンテーブルに示された最大また最小の部分登録ビットパターンに対応して決まる最大または最小の登録ビットパターンを示すバイパスデータを有する場合は、下位の照合段階の照合テーブルが更新されないとバイパスデータが決定されない場合がある。したがって、更新過程においては、上位の照合段階の照合テーブルから下位の照合段階の照合テーブルを更新する際に、バイパスデータの更新が発生する可能性のある照合段階の照合テーブルの更新は、下位の照合段階の照合テーブルの更新を待ってから行う、待ち合わせ方式が採用できる。また、更新過程においては、上位の照合段階の照合テーブルから下位の照合段階の照合テーブルを更新する工程と、下位の照合テーブルの更新により上位の照合テーブルのバイパスデータの更新が発生した場合は、双方向リンクにより上位の照合テーブルの更新を行う工程とを有する、双方向リンク方式を採用することも可能である。

複数の照合装置を備えた分類装置により多段階の照合方法により検索を行うことができる。照合装置は1または複数の照合段階を実行するものであるが、1つの照合段階をそれぞれ実行する複数の照合装置（照合段階）をパイプライン方式

などで接続する分類過程を用いた分類方法、および分類過程を実装した分類装置は、各照合段階の管理および制御が容易であり、本発明の好ましい形態の1つである。複数種類の検索対象のビットストリングを対象とする検索を実行する必要がある場合、照合装置を複数有する1つの分類装置において複数種類の検索を順番に実行することができる。複数の分類過程を有する分類方法、および複数の分類装置を備えた検索装置を提供することはさらに有用である。複数の分類過程および分類装置により分類処理を並列に実行することも可能である。複数種類の検索対象のビットストリングを連結して1つの検索対象のビットストリングとして照合を行うことも可能である。しかしながら、検索対象のビットストリングのデータ長が増えるほど、下流におけるパターンテーブルの数が膨大になる可能性がある。

登録ビットパターンが1つまたは複数の分類結果（ルール）を示している場合は、範囲検索により検索対象のビットストリングが属する範囲が決まれば、その検索対象のビットストリングの分類結果が決まる。複数の分類過程を備えた分類方法においては、複数の分類結果が得られる。したがって、複数の分類結果の論理積を演算し、最終の分類結果を得る論理演算過程を設ける必要がある。この論理演算過程および論理演算手段においては、複数の分類結果の論理積を行列演算し、最終の分類結果を得ることができる。このような論理演算過程（論理演算工程）および手段により、処理速度を向上でき、構成を簡易にできる。複数の分類装置を備えた検索装置においても同様である。

分類装置において、さらに処理速度を向上するのに最も効果的であると考えられるのは、メモリに対してデータを入出力するキャッシュ装置である。そして、パターンテーブル、特にマスクデータおよびアドレステーブルを含む照合テーブルのサイズとキャッシュ装置のキャッシュラインサイズが同一であることが望ましい。さらに、異なる種類の検索を行う分類装置においては、最初の照合テーブルを指定する検索種別情報を受信する手段を設けることが望ましい。キャッシュ装置は、それぞれの検索種別および照合段階の単位でキャッシュラインを割り当て、検索種別および照合段階の単位でキャッシュラインを管理する管理手段を備

えていることが望ましい。メモリの、それぞれの検索種別および照合段階の単位で割り当てられた個別アドレス領域に照合テーブルを記憶しておき、キャッシュ装置のキャッシュラインを、個別アドレス領域単位で割り当てることも可能である。

- 5 ビットストリングの検索においては、全体が同一または上位が同一のビットストリングが供給されることが多い。このため、従来のCPU用のキャッシュシステムのように、メモリ上のデータの局所性を期待したキャッシュシステムを採用するよりは、検索種別および段階に区分けして、その単位で過去の参照テーブルを保存するキャッシュ装置の方が、キャッシュヒット率が高く、検索あるいは照
10 合処理速度の向上に効果がある。

- また、キャッシュ装置としては、照合テーブルがオンキャッシュでなければ照合装置にその旨を通知し、その照合装置の処理を中断して待ち行列に戻す、要求応答型のデータキャッシュも有効である。照合テーブルがオンキャッシュでない
15 ときに、照合装置、データキャッシュおよびメモリが、そのアクセスに占有されてしまい、処理が遅れることを防止できる。パイプライン的に照合装置が接続されている場合は、1つ遅れると、その遅れは後段に影響するからである。また、検索対象のビットストリングに識別情報を付加することにより、入出力の順序が保障される必要はなくなるので、待ち行列で待機させても照合処理には影響がない。検索対象のビットストリングに識別情報を付加して照合する方法は、範囲一
20 致検索において、バックトラックが発生したときにも有効である。

- また、全体が同一の検索対象のビットストリングが供給される頻度も高い。したがって、分類装置の分類結果を記憶した履歴キャッシュ装置を設置することも有効である。検索対象のビットストリングを照合装置に供給する前に、当該分類装置の分類結果と検索対象のビットストリングとを比較する履歴キャッシュを設
25 けることにより、検索あるいは照合処理速度を向上できる。

 ビットストリングの検索あるいは照合を高速で処理することができる本発明は、デジタルデータを取り扱うすべての処理、アプリケーション、装置、システムにおいて有用である。本発明の適用範囲を限定するものではないが、その1つとし

ては、MPEGなどの画像データを示すビットストリームから特定のビットスト
リングを検索したり、膨大な文字データから特定の条件を検索するなどの多様な
アプリケーションが考えられる。高速性が要求されるコンピュータネットワーク
を対象とするデータ管理装置の分野にも有効である。たとえば、分類装置や検索
5 装置の照合あるいは検索により特定されたルートにパケットを送出する機能を備
えた、高速ルーティングが可能なデータ管理装置あるいは管理方法を提供できる。
また、ルータやサーバなどで必要となるパケットフィルタリングSPD検索、Q
oSといった範囲検索を高速で実行できるデータ管理装置を提供できる。さらに、
TCPパケットのリアセンブルのように、データブロックの先頭アドレスおよび
10 末尾アドレスが一致するだけでなく、前後および包含関係になることもあるフラ
グメント化されたデータブロックを再構成するアプリケーションにおいても有効
である。

さらに、本発明の照合方法および照合装置は、大規模なデータベースから有用
なルールあるいはパターンを高速に発見するデータマイニングと呼ばれる分野で
15 も有効である。さらに、本発明の照合方法および照合装置を用いた検索がツリー
の一方方向遷移に限定される必要はない。このような視点はミクロ的なものであり、
マクロ的な視点では双方向ネットワークの遷移先などを検索する方法および装置
も本発明に含まれる。その1つは、状態遷移マシンであり、有限状態オートマト
ンである。したがって、本発明の照合方法により、状態遷移の評価元となるデー
20 タを検索対象のビットストリングとし、複数の状態遷移条件を示すデータを複数
の登録ビットパターンとして照合する工程と、その照合結果によりデータ処理回
路の状態を遷移する工程とを有するデータ処理装置の制御方法を提供できる。

本発明により、状態遷移の評価元となる現状態の検索対象のビットストリング
を、予め登録された複数の状態遷移条件を示す登録ビットパターンと照合する照
25 合装置と、照合装置に対し、検索対象のビットストリングを供給する検索対象供
給手段と、照合装置の出力により状態が遷移するデータ処理回路とを有するデー
タ処理装置を提供できる。このデータ処理装置は遷移状態マシンあるいはオート
マトンプロセッサであり、分類器プロセッサとも称することができる。この場合、
照合装置は、現状態の検索対象のビットストリングが取り得る全ての値と照合す

る全照合手段と、複数の登録ビットパターンを示すためのパターンテーブルであって、先行する状態から得られる照合継続情報により決まる、現状態のパターンテーブルをメモリよりロードするパターンロード手段と、全照合手段の結果と現状態のパターンテーブルより、現状態の検索対象ビットストリングに一致する

5 現状態の登録ビットパターンの有無を少なくとも示す照合結果を出力する判定手段と、照合結果により、現状態のパターンテーブルに対応するアドレステーブルから、現状態の次の状態の前記パターンテーブルのアドレスを含む照合継続情報を出力する出力手段とを備えている。

- 1 つの状態遷移を決定するのに、複数の照合装置を用いた設計も可能であり、
- 10 その場合は、本発明の分類装置が遷移先の検索に用いられる。照合する内部状態、すなわち、状態遷移条件を外部条件や内部条件で変更することが可能であり、検索対象供給手段により、照合装置に供給される照合テーブルが指定されるようにしても良い。データ処理回路によりメモリの照合テーブルが書き換えられるようにしても良い。Q o S の検索などに本発明の状態遷移（オートマトン）プロセッサを用いると、Q o S の検索とプロセッサとしてのアドレッシングが一体となつ
- 15 たアーキテクチャを実現することができる。

図面の簡単な説明

図 1 は、管理装置の例としてルータの概要を示す図である。

- 20 図 2 (a) ~ 図 2 (f) は、一致検索の照合状態を説明する図である。

図 3 は、バイナリビットツリー上の照合状態を示す図である。

図 4 (a) ~ 図 4 (f) は、右倒れ一致型の近似について説明する図である。

図 5 (a) ~ 図 5 (f) は、左倒れ一致型の近似について説明する図である。

図 6 は、バイナリビットツリーをテーブル分割した様子を示す図である。

- 25 図 7 (a) ~ 図 7 (c) は、照合テーブル（部分テーブル）を示す図である。

図 8 (a) および図 8 (b) は、MIN/MAX サーチを説明する図である。

図 9 (a) および図 9 (b) は、部分テーブルで MIN/MAX サーチを行う様子を示す図である。

図 10 は、圧縮テーブルで MIN/MAX サーチを行う様子を示す図である。

図 1 1 (a) および図 1 1 (b) は、バイパス処理を示す図である。

図 1 2 は、照合過程でバックトラックが発生する様子を示す図である。

図 1 3 は、バックトラックが発生した後に、バイパス処理を実行する様子を示す図である。

5 図 1 4 は、検索装置の概要を示す図である。

図 1 5 は、メモリ共用型の分類装置を示す図である。

図 1 6 は、メモリ分離型の分類装置を示す図である。

図 1 7 は、繰り返し型の分類装置を示す図である。

図 1 8 は、照合装置の概要を示す図である。

10 図 1 9 は、照合装置の処理を示すフローチャートである。

図 2 0 は、照合装置における処理の進行を説明する図である。

図 2 1 は、比較ユニットを示す図である。

図 2 2 は、比較器の論理を示す図である。

図 2 3 は、比較ユニットの異なる例を示す図である。

15 図 2 4 は、マスク器を示す図である。

図 2 5 は、マスク器の論理を示す図である。

図 2 6 は、選択回路の一致論理を示す図である。

図 2 7 (a) ~ 図 2 7 (c) は、選択回路の範囲検索 (右倒れ) の論理を示す図である。

20 図 2 8 (a) ~ 図 2 8 (c) は、選択回路の範囲検索 (左倒れ) の論理を示す図である。

図 2 9 (a) ~ 図 2 9 (g) は、比較ユニット、マスク器、選択回路の出力 (右倒れ) を示す図である。

25 図 3 0 (a) ~ 図 3 0 (g) は、比較ユニット、マスク器、選択回路の出力 (左倒れ) を示す図である。

図 3 1 は、コマンドの例を示す図である。

図 3 2 は、バイパス処理用のバスでバックトラックを処理する様子を示す図である。

図 3 3 は、メモリ共有型の分類装置でバックトラックを処理する様子を示す図

である。

図34は、分類装置においてスルーを処理する様子を示す図である。

図35は、異なる方式でスルーを処理する様子を示す図である。

図36は、さらに異なる方式でスルーを処理する様子を示す図である。

5 図37は、さらに異なる方式でスルーを処理する様子を示す図である。

図38は、待ち合わせ方式でエントリを追加する処理を示す図である。

図39は、待ち合わせ方式でパイプラインに沿って処理を進める様子を示す図である。

10 図40は、待ち合わせ方式でエントリを追加する更新ユニットの機能を示す図である。

図41は、待機管理ユニットの機能を示す図である。

図42は、双方向リンク方式でエントリを追加する処理を示す図である。

図43は、双方向リンク方式でパイプラインに沿って処理を進める様子を示す図である。

15 図44は、双方向リンク方式でエントリを追加する更新ユニットの機能を示す図である。

図45は、待ち合わせ方式でエントリを削除する処理を示す図である。

図46は、待ち合わせ方式でエントリを削除する更新ユニットの機能を示す図である（コマンドがDELのとき）。

20 図47は、待ち合わせ方式でエントリを削除する更新ユニットの機能を示す図である（コマンドがDELUPDTのとき）。

図48は、待ち合わせ方式でコマンドがDELRDBPおよびDELWRBPの処理を示す図である。

25 図49は、双方向リンク方式でエントリを削除する更新ユニットの機能を示す図である（コマンドがDELのとき）。

図50は、双方向リンク方式でエントリを削除する更新ユニットの機能を示す図である（コマンドがDELUPDTのとき）。

図51は、双方向リンク方式でコマンドがDELRDBPおよびDELWRBPの処理を示す図である。

図 5 2 は、データキャッシュを備えた分類装置を示す図である。

図 5 3 は、データキャッシュのラインサイズを示す図である。

図 5 4 は、データキャッシュの管理を示す図である。

図 5 5 は、メモリの管理を示す図である。

5 図 5 6 は、空テーブル管理キューの応答を示す図である。

図 5 7 は、SDRAM 共用型履歴キャッシュシステムを備えた分類装置を示す図である。

図 5 8 は、SDRAM 共用型履歴キャッシュシステムの概要を示す図である。

図 5 9 は、履歴SRAMキャッシュ付き分類装置を示す図である。

10 図 6 0 は、履歴SRAMキャッシュの概要を示す図である。

図 6 1 は、行列演算の一例を示す図である。

図 6 2 は、行列演算の左行列を示す図である。

図 6 3 は、LSI 化に適して行列演算の概要を示す図である。

図 6 4 は、AND 化器のハードウェアを示す図である。

15 図 6 5 は、下位テーブルを省略する様子を示した図である。

図 6 6 は、要求応答方式のメモリアクセスを採用した照合装置の概要を示す図である。

図 6 7 は、要求応答方式のメモリアクセスを採用した照合装置の異なる例を示す図である。

20 図 6 8 (a) ~ 図 6 8 (c) は、照合テーブルの改良型を示す図である。

図 6 9 は、非リニアメモリアドレッシングを示す図である。

図 7 0 は、単純状態遷移を示す図である。

図 7 1 は、イベント連動型単純状態遷移を示す図である。

図 7 2 (a) ~ 図 7 2 (c) は、状態遷移の概要を示す図である。

25 図 7 3 (a) および図 7 3 (b) は、ネットワーク状の状態遷移の概要を示す図である。

図 7 4 は、分類装置を採用したプロセッサの概要を示す図である。

図 7 5 (a) および図 7 5 (b) は、QoS に適応する概要を示す図である。

図 7 6 は、検索ツリーとクラスツリーの対応を示す図である。

図 7 7 は、従来の CPU による検索処理の概要を示す図である。

図 7 8 は、分類装置を採用したプロセッサによる検索処理の概要を示す図である。

5 発明を実施するための最良の形態

以下では、本発明に係るビットストリングの検索装置をルーティングテーブル検索に使用したルータ（ルーティング装置）について説明する。図 1 に示すルータ 1 は、ネットワーク上のパケットデータを管理する装置であり、パケット ϕp の入出力を制御および管理するパケット管理機能 3 と、パケット ϕp の転送先を判断するためのルーティング検索機能 4 と、パケット ϕp を転送する条件を判断する条件判断機能 9 とを備えている。ルーティング検索機能 4 は、入力されたパケットの経路を示すエントリ（登録ビットパターン）を複数備えたルーティングテーブルが記憶されたメモリ（SDRAM）11 と、検索装置 10 と、検索装置 10 に対して検索種別を設定するなどの機能を備えた制御ユニット 12 とを備えている。ルーティング検索機能 4 では、パケット ϕp に含まれる検索対象のビットストリング（被検索ビットストリング） ϕo 、たとえば、IP ヘッダの受信先アドレス（DIP）をメモリ 11 に記録されたルーティングテーブルの各エントリと一致検索を行う。そして、その結果 ϕr に基づき、パケット管理機能 3 が一致したエントリに設定された経路（インタフェース）へパケットデータを送出する。

条件判断機能 9 は、さらに幾つかの機能に分割されている。本例の条件判断機能 9 は、ネットワーク上のセキュリティーサービスを提供する IP セキュリティー（IPsec）機能 8 と、ネットワーク上のクオリティーサービスを提供する QoS 機能 7 と、ファイアウォール機能 6 を備えている。これらの条件を判断する機能の多くは、パケット ϕp を管理する IP ヘッダのビットストリングを検索対象のビットストリング ϕo として、その検索対象のビットストリング ϕo が属する範囲を検索することが要求される。IPsec（SPD）機能 8 では、制御ユニット 8c の制御のもと、検索対象として供給されたビットストリング ϕo と、メモリ 8m に記憶された SPD テーブルとを検索装置 8s により照合する。

QoS機能7では、QoS制御ユニット7cの制御のもと、検索対象のビットストリング ϕo と、メモリ7mに記録されたフロー検索テーブルとを検索装置7sにより照合する。さらに、ファイアウォール機能6では、制御ユニット6cの制御のもと、検索対象のビットストリング ϕo と、メモリ6mに記録されたパケットフィルタリングテーブルとを検索装置6sにより照合する。

パケット管理機能3は、ネットワークを介してパケット ϕp を受信および／または送信可能な1つまたは複数のパケット入出力部2aと、受信したパケット ϕp をキューイングするメモリ2bとを備えている。キューイングされたパケット ϕp は、ルーティング検索機能4でルーティング先が決まる。さらに、SPD機能8、QoS機能7およびファイアウォール機能6などにおける処理が決定された後に、それらの機能で決定された適切なタイミングで、パケット ϕp は、パケット管理部3の何れかのパケット入出力部2aから適切なルートあるいはネクストホップに送出される。また、SPD機能8などの照合結果によってはキューイングされたパケット ϕp が出力されずに破棄されることもある。

本発明の照合方法を採用した検索装置は、これらの各機能4、6、7および8の検索装置10、6s、7sおよび8sとして適用可能であり、IPv6に基づいて検索対象のビットストリング ϕo のビット長が増した状況したでも、高速でサーチし、その結果 ϕr をパケット管理部3に返すことができる。条件判定機能9に含まれる機能6、7および8の検索装置6s、7sおよび8sにおいては、検索対象のビットストリング ϕo が属するルールを見出すために、範囲一致検索あるいは照合が実行される。一方、ルーティング検索機能4においては、ルートを特定するために完全一致検索または最長一致検索が実行される。ルーティング検索機能4において、プレフィックス以降の最長一致の結果を得る最長一致検索は、プレフィックス以降のすべてのビットを最小側と最大側に展開した範囲一致検索に還元される。なお、本明細書において、バイナリビットツリーの「0」の方向を最小側(MIN)と呼び、「1」の方向を最大側(MAX)と呼ぶ。

検索装置の基本原理

以降においては、本発明に係る検索装置の概要を説明する。本発明の検索装置

は複数種類の検索を単一のハードウェアで行えるものであり、いずれの機能の検索装置としてもほとんど構成を変えずに適用可能である。したがって、以降においては検索装置 10 を、完全一致検索および範囲一致検索が可能な装置として説明するが、範囲検索も同様のハードウェアで実行できる。

- 5 検索装置 10 は、バイナリビットツリーを構成するエントリ群と被検索キー（検索対象キー、検索対象のビットストリング、以降においてはターゲットキーあるいはキーと称することもある）の間の大小判定アルゴリズムを用い、エントリ（登録ビットパターン）群が表現する範囲（ルール）のどこに対象キーが属するかを判定するハードウェアである。バイナリビットツリーに基づく照合を実行
- 10 するアルゴリズムをハードウェアで実現するにあたり、バイナリビットツリーを短いビット長のテーブルに分け、多段階で照合を行う。各照合段階では、各段のテーブル内の大小判定を並列ビット照合により行う。多段階の照合にすることにより、検索対象となるビットストリングのビット長の制限はなくなり、汎用的な利用が見込まれる。また、多段階の照合にすることにより、各段階で比較する
- 15 ビット長を限定でき、予めすべてのエントリが登録されているものとして並列に照合しても処理時間への影響はなく、ハードウェアが増大することも無い。そして、各段に登録されているエントリが含まれたパターンテーブルを示すデータ（以降ではマスクデータ）は圧縮することが可能であり、バイナリビットツリー作成（登録・削除）コストが軽減され、多段テーブルによる圧縮効果と相まって
- 20 必要メモリ量も抑えられる。

- 照合過程の各段階における並列ビット照合は逐次実行プロセッサ上のソフトウェアでは実現不可能であり、並列処理ハードウェアあるいは並列実行型のプロセッサで実行することになる。並列ビット照合に必要とされるハードウェアは簡易なものであり、並列実行型のプロセッサを用いずに、経済的、コンパクトに実
- 25 現することができる。

バイナリビットツリーによる同値および大小の判定

以下で述べる範囲検索において、右倒れ型近似ロジック、左倒れ型近似ロジックはハードウェア設計時にどちらかを選んで設計する。どちらを選んでも等価な

範囲検索が可能である。両方式を切り替えられるよう設計することも可能である。一度に両方の近似で検索することも可能である。本明細書では、特に記載しない限り、右倒れ型近似ロジックに基づき設計された装置により説明する。

図 2 (a) から図 2 (f) に、バイナリビットツリー上での完全一致検索のロジックを示してある。ツリーの 1 段が 1 ビットに相当するバイナリビットツリーにおいては、上位から 1 ビットずつ被検索キーとエントリとの間で照合していく。この上位とは照合を開始するビットであり、必ずしも最上位のビット (MSB) でなくても良く、最下位のビット (LSB) であっても良い。バイナリビットツリーのあるノードから次ノードへの接続は「両方向あり」、「0 方向のみあり」および「1 方向のみあり」の 3 通りしかない。1 ノードが 1 エントリを意味するバイナリツリー (2 分木) とは異なり、バイナリビットツリーではノードはエントリの 1 ビットを意味するに過ぎない。エントリは全て等ビット長なのでツリー上でのノード段数も等しく、上位ノードからの接続が存在するノードを辿っていく途中で下位ノードに対する接続が「両方向なし」となることはありえない。

本明細書において、図 2 (a) および図 2 (b) に示すように、「検索対象キー 21 のビットとエントリ 22 のビットが一致」となるエントリは「照合候補」と定義する。また、図 2 (c) および図 2 (d) に示すように、「検索対象キー 21 のビットとエントリ 22 のビットとが不一致」となるエントリは「照合失格」と定義する。図 2 (e) および図 2 (f) に示すように、ノード接続が両方向である場合は、検索対象キー 21 のビットとエントリ 22 a のビットが一致」となるエントリ 22 a は照合候補で、「検索対象キー 21 のビットとエントリ 22 b のビットが不一致」となるエントリ 22 b は照合失格である。

完全一致ロジックにおいては、各ビット (各ノード) で照合をおこない、照合候補が存在する間、照合候補の次ノードを辿り、照合を継続する。照合候補が無くなればそこで照合は終了し、検索ミスヒットという最終照合結果が得られる。最終ビット (最終ノード) の照合完了後に照合候補があれば (状態が照合継続であれば)、それに対応するエントリが検索ヒットである。

図 3 は、バイナリビットツリーにおける完全一致検索の照合ロジックを表すものである。バイナリビットツリーに登録された 4 ビットのエントリ群 22 (値 0,

2, 8, 15) に対し、検索対象キー 21a として値 8 (「1000」) を投入し、値 8 に対応するエントリが完全一致検索ヒットと判定される例と、検索対象キー 21b として値 4 (「0100」) を投入し、検索ミスヒットと判定される例を示してある。最初のビットであるノード A で検索対象キー 21a は 1 方向に進む。

- 5 値 0 および 2 のエントリ 22 は 0 方向、値 8 および 15 のエントリ 22 は 1 方向なのでエントリ 22 は両方向に延び、前者のエントリ 22 は検索対象キー 21 のビットと異なるので照合失格、後者のエントリ 22 は検索対象キー 21 のビットと一致するので照合継続である。そして、このノード A での照合後に照合候補エントリが存在するため、照合継続である。

- 10 次のノード E では、検索対象キー 21a は 0 方向に進み、値 8 のエントリ 22 が 0 方向、値 15 のエントリ 22 が 1 方向に延びている。したがって、ノード E でも照合候補が存在するので照合継続である。ノード F では、検索対象キー 21a は 0 方向に進み、エントリ 22 は片方向であるが、その値 8 のエントリ 22 が検索対象キー 21a と同じく 0 方向なので照合候補が存在し、照合継続である。
- 15 ノード G でも同様であり、最終ノードである。したがって、最終ノード G の照合後に照合候補があるので、完全一致検索は値 8 のエントリ 22 にヒットし終了する。

- 検索対象キー 21b では、ノード A では照合候補があるため照合が継続する。ノード B では、検索対象キー 21b が 1 方向、値 0 および 2 のエントリ 22 は片
- 20 方向で 0 方向に延びている。したがって、照合継続となっていた値 0 および 2 のエントリ 22 は照合失格となり、このノードでの照合後に照合候補が残らない。このため、検索ミスヒットとなる。

- 範囲検索でも完全一致検索同様にバイナリビットツリーを上位から 1 ビットずつ検索対象キーとエントリとの間で照合していくが、範囲検索の場合は「照合候補」に加え、「近似候補」および「近似失格」の概念がある。完全一致検索での
- 25 「照合失格」が「近似候補」と「近似失格」に区別されることになる。また、ロジック上の必要性から「近似仮定」も定義する。

右倒れ近似 (MAXサーチ)

図4 (a) から図4 (f) に右倒れ近似の類型を示してある。図4 (a) および図4 (b) に示すように、「検索対象キー21のビットとエントリ22のビットが一致」となるエントリは「照合候補」と定義する（これは完全一致検索と同様である）。この照合候補がある間、右倒れ型近似ロジックによる照合は継続する。また、図4 (e) および図4 (f) に示すように、「エントリ22が両方向あり」の場合も、「検索対象キー21のビットとエントリ22のビットが一致」である側のエントリ22は照合候補となり、照合継続である。

一方、図4 (c) および図4 (d) に示すように、「検索対象キー21のビットとエントリ22のビットが不一致」となるエントリは照合候補から外れる。完全一致検索ではいずれの場合も「照合失格」としていたが、右倒れ型近似の場合、図4 (c) のように「検索対象キー21のビットが1でエントリ22のビットが0」となるエントリ22を「近似候補」と定義する。逆に、図4 (d) のように「検索対象キー21のビットが0でエントリ22のビットが1」となるエントリ22を「近似失格」と定義する。近似ロジックでは、最新の近似候補を「近似仮定」として保持しておく。実際に「近似仮定」として保存が必要な値は、近似候補の指す次ノード位置である。近似候補でも近似失格でも、図4 (c) および図4 (d) に示すようにエントリ22が片方向で照合候補がなければ、照合過程は終了する。

図4 (e) および図4 (f) に示すように、エントリ22が「両方向あり」の場合は、「検索対象キー21のビットとエントリ22のビットが不一致」となる側のエントリ22も同様に、定義に従い近似失格あるいは近似候補となる。「両方向あり」なので「検索対象キー21のビットとエントリ22のビットが一致」側のエントリ22も存在し、それが照合候補となるので照合は継続する。それと共に、「近似候補」となるエントリ22、すなわち、図4 (e) の「近似候補」のエントリ22が「近似仮定」となり、それまでの「近似仮定」の値は更新される。

最終ビットの後に照合継続（照合候補あり）ならば、そのエントリに完全一致したことになる。途中（あるいは最終ビット）で照合終了となった場合、保持し

てある近似仮定（最新の近似候補）を見て近似ヒットがありえるかどうかを判定する。近似仮定がなければ検索ミスヒットである。範囲検索の意味的には、定義された全ての範囲の外である。近似仮定があれば近似ヒットである。しかし、近似仮定が途中のノードを指す場合は、そのノードの下に複数のエントリ 2 2 が存在する可能性があるため、近似仮定のエントリ群 2 2 から近似決定をおこなわねばならない。この過程は、近似照合とは別のロジックで処理できる。右倒れ近似においては、以下で最大値検索（MAXサーチ）として説明している処理を行い、検索対象キー 2 1 の値以下の値のエントリの中から、最大の値のエントリ（検索対象キーに近い最大のエントリ）を選択して近似エントリとして確定する。

- 10 図 3 を、バイナリビットツリーにおける右倒れ型近似ロジックによる範囲検索を表すものとする。以下のようになる。値 4 の検索対象キー 2 1 b については、ノード A で 0 方向の値 0 または 2 のエントリ 2 2 は照合候補、1 方向の値 8 または 1 5 のエントリ 2 2 は近似失格となる。ノード A では、照合候補が存在するため照合継続となる。次にノード B では、0 方向の値 0 または 2 のエントリが近似候補となり、近似候補なので近似仮定（最新の近似候補を保持）の更新を行い、
- 15 近似候補の指す次ノードであるノード C を近似仮定として保持する。

- ノード B の照合後に照合候補がなく、照合過程はいったん終了する。近似仮定があるので、近似ヒットである。従って、近似仮定となるエントリ群から MAX サーチにより近似確定をおこなう。この場合、値 0 と値 2 のエントリ 2 2 から値
- 20 2 のエントリを選択することになる。図 3 に示したように、検索対象キー 2 1 b は $2 \leq \text{キー値} \leq 7$ の範囲の値であれば、同じ結果である値 2 のエントリ 2 2 を得ることができる。つまり、値 2 のエントリ 2 2 は（2～7）という範囲を意味するエントリとなっている。このため右倒れ近似が範囲検索になるのである。

- 次に、値 8 の検索対象キー 2 1 a においては、ノード A では、0 方向のエントリ 2 2（値 0 と 2 が属する）が近似候補となり、近似仮定（ノード B を指す）が更新される。1 方向のエントリ 2 2（値 8 と 1 5 が属する）が照合候補となり、照合候補も存在するため照合継続となる。ノード E では、ビット値 0 方向のエントリ 2 2 が照合継続、ビット値 1 方向のエントリが近似失格となる。近似候補はないため、近似仮定は更新されない。ノード F 以降は単方向で照合継続となり、
- 25

近似候補が発生しないため、近似仮定は更新されない。そして、最終ビット後に照合継続（照合候補あり）なので、値8のエントリ22に完全一致するという最終照合結果が得られる。

- さらに、値14の検索対象キー21cを投入した場合を考える。ノードAでは、
- 5 ビット値0方向のエントリ22（値0と2が属する）が近似候補となるので、近似仮定（ノードBを指す）が更新される。また、ビット値1方向のエントリ（値8と15）が照合候補となり照合継続となる。ノードEでは、ビット値0方向の値8のエントリ22が近似候補となり、近似仮定は更新される（ノードFを指す）。また、1方向の値15のエントリ22が照合候補となるので、照合継続と
- 10 なる。ノードHでは、単方向で値15のエントリ22が照合候補となり、近似候補が発生しないので、近似仮定は更新しないで、照合継続となる。ノードIでは、単方向で照合失格となる。従って、いったん照合過程は終了する。

- このとき、近似仮定はノードEでの更新でノードFを指している。近似確定のため、ノードF以下をMAXサーチし、値8に対応するエントリ22が近似ヒッ
- 15 トと決定される。したがって、この検索対象キー21は、値8のエントリ22がカバーする範囲（8～14）に属することが判明し、範囲一致検索が実行されたことになる。

左倒れ近似（MINサーチ）

- 20 図5（a）から図5（f）に、左倒れ近似における「照合候補」「近似候補」「近似失格」の定義を示してある。右倒れ近似と「近似候補」「近似失格」のビットが入れ替わっただけである。照合ロジックも、上記の入れ替わり以外は右倒れ近似と同様である。ただし、右倒れ型近似では近似確定するため近似仮定に対するMAXサーチをおこなうのに対し、左倒れ近似の場合、近似確定のために
- 25 近似仮定に対するMINサーチをおこなう点が異なる。MINサーチでは、検索対象キー21の値以上の値のエントリの中から、最小の値のエントリ（検索対象キーに近い最小のエントリ）を選択して近似エントリとして確定する。

最終ビットまで「エントリのビットと検索対象キーのビットが一致」であった場合、つまり最終ビットの照合後に照合継続であった場合、これも範囲検索とし

てはヒットとする。つまり、右倒れ型近似による範囲検索とは、完全一致ヒットと近似一致ヒットを区別せず、「キー \geq エントリ」となる最大エントリを検索するものである。ただし、検索の用途によっては「 $=$ 」を含まない、「キー $>$ エントリ」の右倒れ近似とすることもできる。また、「 $=$ 」一致なのか「 $>$ 」一致なのかを区別した結果とすることもできる。

範囲検索におけるエントリデータの形式

右倒れ型近似による範囲検索では、領域は右倒れ近似ロジックに適合するよう正規化されエントリ登録される。正規化とは、複数のルールが与えられたときに、相互に重複しない新たな範囲であって、その新たな範囲に属するルールの重複を許した、新たな範囲を設定することを言う。そして、検索対象ビットストリングが属する範囲を判断することにより検索対象ビットストリングに適用されるルールを発見する。また、検索対象のビットストリングが属する新たな範囲に複数のルールがあるときは、優先度の高いルールに基づき検索対象のビットストリングが属するパケットを管理するという方法でも用いられる。

さらに説明すると、ルールを正規化するとは、照合処理のために範囲の数値表現をFROM/TO表現からFROM/NEXT表現にすることを意味する。すなわち、ルールと範囲の関係を、ルールの重複は許し、範囲の重複は許さないで分割した領域表現に変換する。たとえば、 P_i を領域中の値とし、範囲 $[P_a, P_a']$ と範囲 $[P_x, P_x']$ が与えられたときに、 $P_a < P_x < P_a'$ の関係があるときは、 $[P_a, P_x - 1]$ 、 $[P_x, P_a']$ 、 $[P_a' + 1, P_x']$ の3つの新たな範囲を定義する。このような正規化により、新たな範囲のいずれかに属することが分かれば、それに対応するルール（優先するルール）を一義的に決定することができる。すなわち、範囲 $[P_a, P_a']$ は、 $[P_a, P_x - 1]$ および $[P_x, P_a']$ の2つの新たな範囲に分割される。従って、 m 個のルールが与えられたときは、領域数（新たな範囲の数）を最大で $2m + 1$ に分割することにより、重複することのない新たな範囲を設定することができる。最大となる条件は、すべてのルールの範囲のFROM/TOがユニークな場合であり、実際は、重複することが多いために新たな範囲の数はかなり少ない。

- また、この新たな範囲には、ルールが与えられていない範囲も、ルールが与えられていない範囲として意味を持つことになる。したがって、FROM/TOによる範囲は[F r o m, *] および[N e x t, *] という範囲に変換する必要がある。つまり、FROM/TOのペアで1範囲を表現していたものが、F r o m値/N e x t値それぞれが独立領域を表現することになる。F r o m値は元のルール範囲のF r o m値と同値であり、ルール範囲の開始位置（左端点）を示す。N e x t値はルール範囲のT o（範囲終了位置）+ 1であり、ルール範囲の右側の範囲外領域の開始位置（左端点）である。これにより、F r o m値/N e x t値は新たな範囲の端点として独立した存在となり、新たな範囲は常に左端点（下位のビットストリング）の1点で表現され、正規化された定義となる。すなわち、新たな範囲の右端点はN e x t領域左端点- 1で暗黙に表現される。なお、ルール領域の右端点を中心に考えて、新たな範囲を上位のビットストリングの1点で表現することも可能である。その場合は、以下の説明を上位と下位を逆転させて考えれば良い。
- 15 正規化により、FROM/TOのルールエントリは、FROM/NEXTのエントリに変換される。この際、デフォルトのエントリとして[0, NA]を追加する。したがって、ルール数がnであれば、エントリはデフォルトも含めn+1エントリとなる。正規化する処理は、範囲により照合されるものがルールに限定されることはなく、範囲に対応して出力する値が特定される検索であれば全ての
- 20 場合に応用することができる。

この正規化は領域を表す2値（[FROM, TO]）のうちTO（大値）側の補正ということができる。例えば[a, b]で表される領域Rは値「a」のエントリAと値「b+1」のエントリBとして登録される。エントリAは領域Rを表現し、エントリBは領域S[b+1, ∞]を表現する。エントリAに右倒れ型近似ヒットすることは、領域Rに範囲検索ヒットすることである。エントリBに近似ヒットすることは、領域Sにヒット、つまり範囲検索ミスヒット（領域R外）を意味する。近似ミスヒットは領域Q[0, a-1]、つまり範囲検索ミスヒット（領域R外）を意味する。

一点を表す領域U[c, c]も定義可能であり、値「c」のエントリCと値

- 「 $c + 1$ 」のエントリDとして登録される。エントリCは領域Uを表現し、エントリDは領域V [$c + 1, \infty$] を表現する。値「 c 」と値「 $c + 1$ 」の隣接するエントリが登録されるため、値「 c 」であるエントリCへのヒットは完全一致ヒットでしかありえず、点領域Uに範囲検索ヒットを意味する。エントリDへの
- 5 近似ヒットは領域Vにヒット、つまり範囲検索ミスヒット（領域U外）である。近似ミスヒットは領域T [$0, c - 1$]、つまり範囲検索ミスヒット（領域U外）を意味する。

左倒れ型近似でも領域の正規化はおこなわれる。ただし、右倒れ型とは逆にF ROM（小値）側を補正する。

- 10 エントリと集合の関係を、エントリIは集合SET (I) [i, ∞] を意味すると定義する。このとき、領域R [a, b] は、値「 a 」のエントリAすなわち集合SET (A) [a, ∞] と、値「 $b + 1$ 」のエントリBすなわち集合SET (B) [$b + 1, \infty$] との差集合 (SET (A) - SET (B)) で表せる。エントリAに右倒れ近似ヒットするということは、エントリAにヒットし、エントリBにミスヒットということである。
- 15

照合方法

- 図6に分割テーブルを示してある。多ビットのバイナリビットツリー24は、多段階または多階層に分けて、さらに各段階を適当な数のエントリが表れるビット幅に分割することにより、パターンテーブル化できる。一定の短ビット長を単位とする分割テーブルまたは部分テーブル（パターンテーブル）25を規定することにより、多ビットのバイナリビットツリーを、上位の部分テーブル25から指定される各段階の部分テーブル25の組み合わせにより表現できる。固定ビット数の部分テーブルにすることで、並列処理対象エントリは有限固定となり、部分テーブル25に含まれるエントリをハードウェアにより並列照合することができる。さらには、部分テーブル25に含まれる最大エントリ数は有限固定になるので、実際に部分テーブル25に含まれるエントリの有無にかかわらず、検索対象キーが部分テーブル25に含まれ得るエントリのいずれと一致するかを、ある部分テーブル25が選択されると自動的にハードウェアで照合することも可能と
- 20
- 25

なる。

図 6 では、3 ビットの部分テーブル 25 によりバイナリビットツリー 24 を表現している。各部分テーブル 25 に収容される部分エントリ 26 の最大値は、値 0 ～ 7 の 8 エントリである。ハードウェア実装としては、メモリのワードのビット長や検索キーのビット長などを考慮し適切なテーブルビット長を決定する。

5 テーブルビット長が n ビットになると並列処理対象であるテーブル内のエントリ数が 2^n になるため、一般的には 4 ビットテーブル程度が適切である。

部分テーブル化することにより、各段の部分テーブルでの検索は有限固定エントリ数でのサーチになる。また、検索対象キーが固定ビット幅であれば、エントリを示すバイナリビットツリー 24 は固定長となり、それを表現する部分テーブル 25 の段数は予め定まった有限固定段数となる。このため、一致検索においてはバックトラックの発生しないバイナリビットツリー検索となる。したがって、検索に要するワースト時間はエントリ数にかかわらず一定時間となる。

10

図 7 (a) に、バイナリビットツリー 24 が部分テーブル化されている論理イメージを示す。図 7 (b) は、部分テーブル (パターンテーブル) 25 を汎用テーブル 27 として構成した様子を示してある。この汎用テーブル 27 では、1 エントリはビットパターンの値 27 a と、それに対応する下位テーブル 27 へのポインタ 27 b で構成され、それが最大 n 個あるテーブル構造となる。さらに、図 7 (c) に、汎用テーブル 27 を圧縮したテーブル 28 で部分テーブル 25 を

15 実現した様子を示してある。この圧縮テーブル 28 は、エントリの有効無効を示すビットフラグ (マスクデータ) 28 a と下位へのポインタ 28 b で 1 エントリが構成される。下位の圧縮テーブル 28 へ続くポインタ 28 b は、圧縮テーブル 28 が記憶されている RAM のアドレスなどになり、アドレステーブルを生成する。

25 汎用テーブル 27 と圧縮テーブル 28 の違いは、汎用テーブル 27 では登録ごとにエントリにビットパターンをセットしていくのに対し、圧縮テーブル 28 では、テーブルは固定ビット幅のテーブルであり、ビットパターンを省略している点である。部分テーブル 25 が固定ビット幅の場合、ビット幅に対応して上限エントリ数が決まる。したがって、例えば 4 ビットテーブルの場合、エントリ No

0はビットパターン0000、エントリNo1はビットパターン0001というようにエントリNoとビットパターンの関係を一対一に固定できる。したがって、アクティブなエントリNoを示すことだけで、部分テーブル25に含まれる部分エントリ26を指定することができる。エントリNoを示すデータはいくつか考えられるが、図7(c)に示すようにエントリNoに対応したビット位置にオンオフのフラグとなるビットを配置したビットパターン(マスクパターン)28aはハードウェアで利用しやすい形態の情報である。また、マスクパターン28aにより部分エントリの有無を示すことにより、実際の部分エントリのビットパターンを登録する処理および登録するためのハードウェアおよびソフトウェアを省略できる。

この部分テーブル25をマスクパターンを用いた圧縮テーブル28とした最適化により、部分テーブル25をそれに対応する検索対象キーの部分(部分対象キー)と照合するハードウェアが単純化される。4ビットの部分テーブル25であれば、4ビットの部分エントリ、すなわち、値0から15までの部分エントリが取りうる値と部分対象キーとの照合を並列に行うハードウェアを採用することにより、照合作業およびハードウェアが単純化される。すなわち、全エントリハードウェア並列照合が可能となり、それが単純化される。また、エントリNoとビットパターンの関係が固定でソート済なので、エントリ(部分エントリ)の追加削除は、基本的にはマスクデータ28aを更新するだけで実現できる。そして、テーブル内でのMIN/MAXサーチはビットパターンの比較なしにエントリNoのみでおこなえる。したがって、範囲一致検索において、近似候補となる部分テーブル25までバックトラックが発生したとしても、その後は、エントリが有効か無効かのフラグ(マスクデータ)28aの参照だけでテーブル内におけるMIN/MAXサーチは終わることができるので、実質的にはバックトラックが発生しない照合方法ということも可能である。

MIN/MAXサーチとそのバイパス

MIN/MAXサーチの原理

バイナリビットツリーにおけるMIN/MAXサーチの原理は以下の通りであ

る。図8(a)に、右倒れ近似におけるMAXサーチの概要を示してある。ビット位置(ノード)A1にて、検索対象キー21に対してエントリ22がなくなり、右倒れ型範囲検索の照合候補がなくなる。そのため、照合は途中で終了し、近似仮定エントリに対するMAXサーチを開始する。エントリ22が片方向しかない場合はその方向に辿る。ノードA2においては、0方向にしかエントリ22がないので、それを辿る。次のノードA3においては、エントリ22が0および1の両方向にあるので、大きい方向($0 < 1$ なので1の方向)を優先して辿る。各ノードに対して同様の処理を行い、最下段ビット位置A6まで辿り着き、得られるエントリ22がMAXエントリ(検索対象キーの値よりも小さくて最大の値をとるエントリであり、検索対象キーに近い最大のエントリ)である。

図8(b)は、左倒れ近似におけるMINサーチの概要を示してある。左倒れ型範囲検索の場合、同種のロジックで小さな方向(0の方向)を優先して辿り、MINエントリ(検索対象キーの値よりも大きくて最小の値をとるエントリであり、検索対象キーに近い最小のエントリ)を取得するMINサーチをおこなう。

テーブルにおけるMIN/MAXサーチ

バイナリビットツリーをテーブル化すると、MIN/MAXサーチのロジックもテーブル対応にしなければならない。図9(a)に汎用テーブルでのMIN/MAXサーチのうち、右倒れ型近似におけるMAXサーチを示してある。テーブルA1にて照合失格(キー値21と同値のエントリ22が無い状態)となる。テーブル内で「キー値>エントリ値」となるエントリの中から、最大値であるエントリ(「キー値>エントリ値」の中でエントリ値が最もキー値に近いエントリ)を選択し、近似候補エントリ22xとする。近似候補22xの指す次テーブルが近似仮定エントリ22yである。そして、その近似仮定エントリ22y以下に対するMAXサーチを開始する。

まず、近似仮定エントリ22yのテーブルであるテーブルA2に移る。テーブルA2内のエントリを調べ、最大値をもつエントリを選択する。そのエントリの次段テーブルに移り、以下、最大値を持つエントリ選択と下段への移動を繰り返し、最下段まで到達する。最下段でも最大値エントリを選択し、最大値エントリ

の登録データに辿りつき、MAXエントリを出力する。図9（b）は、左倒れ型範囲検索の場合であり、同種のロジックでMINサーチをおこなう。

図10に、圧縮テーブル28のMIN/MAXサーチのロジックのうち、右倒れ型近似に対応したMAXサーチのプロセスを示してある。まず、圧縮テーブル28ではマスクデータ28aにおいてエントリNoとビットパターンとの関係が固定であり、ソート済の状態に等しいので、テーブル内での最小値/最大値エントリの選択はビットパターンの比較なしに、マスクデータ28aの対応するエントリNoにフラグを立てるだけで行うことができる。つまり、有効無効フラグの参照だけでテーブル内における最小値/最大値エントリの選択は終わる。そして、追加したエントリNoに対応するアドレステーブル28bのレコードにそのエントリNoに続く圧縮テーブル28のアドレスを記録することにより圧縮テーブル28をリストで繋ぐことが可能となり、リニアアドレス上のテーブルでなくてもバイナリビットツリーの多段階検索が可能となる。

右倒れ型範囲検索におけるMAXサーチを説明する。先の照合段階において、テーブルA1のマスクデータ28aにおいて照合候補がなくなり、照合が終了すると、次の現段階においては、近似仮定エントリに対するMAXサーチを開始する。近似候補の選択は、テーブルA1のマスクデータ28aの中で「キー値>エントリ値」となるエントリのうち最大値をもつエントリ（最大のエントリ）を選択することでおこなう。ビットマップによる圧縮テーブルなので、「エントリNo=エントリ値」であり、上記条件に合致するエントリはキー値エントリから左にみていった最初1が立っているエントリが近似候補22xである。この例では照合が終了したのと同じテーブルA1に近似候補エントリ22xが存在する。近似候補エントリ22xの指す次テーブルが近似仮定エントリ22yである。

これにより、次の照合段階では、近似仮定エントリ22yであるテーブルA2に移動する。テーブルA2内で最大値をもつエントリを選択する。「エントリNo=エントリ値」なので、右端からみて最初に1の立つエントリが最大値エントリ22mである。最大値エントリ22mの指す次段テーブルA3に移り、エントリ選択と下段への移動を繰り返し、最下段エントリまで到達する。最下段でも同様に最大値エントリ22mを選択し、最下段での最大値エントリ22mの指す

データに到達する。これにより最大のエントリを得ることができる。左倒れ型範囲検索では、同種のロジックでMINサーチを行う。

MIN/MAXサーチのバイパス

- 5 圧縮テーブル28ではエントリNoとビットパターンの関係が固定でソート済なので、MIN/MAXサーチはビットパターンの比較なしにエントリNoのみでおこなえる。有効無効フラグの参照だけでテーブル内におけるMIN/MAXサーチは終わる。しかしながら、エントリの登録・削除がない限り、下位の部分
- 10 テーブル28を辿って到達する最終的な最大のエントリあるいは最小のエントリは不変である。したがって、毎回、MIN/MAXサーチをするのは無駄であり、図11(a)および図11(b)に示すように、各テーブルにバイパスTAG
- (BP)を設け、そのテーブル28以下でMIN/MAXサーチした結果、すなわち、最大なエントリあるいは最小なエントリを記録しておくことによりMIN/MAXサーチを省略することが可能となる。これにより、最大のエントリ、あ
- 15 るいは最小のエントリを検索するプロセスは、近似仮定した圧縮テーブル28にアクセスするだけのプロセスに代わり、検索時間は大幅に短縮される。

- したがって、近似仮定したエントリを近似確定させる際、近似仮定エントリ22yの指すアドレステーブル28bのアドレス（以降においてはTAG（次段
- 20 テーブルを指すTAG）と称することがある）を使い、次段テーブルのバイパスTAG（BP）により近似確定する最大または最小のエントリを決定することができる。範囲検索の論理として右倒れ型でおこなう場合はバイパスBPには最大のエントリが記録され、左倒れ型でおこなう場合はバイパスBPには最少のエン
- トリが記録される。複数の範囲検索系があった場合、照合のロジック構成としては右倒れ型か左倒れ型かのいずれかに統一する。バイパスBPに両値を記録することにより、テーブルに記録する情報量の増加、照合用のロジックの追加、エン
- 25 トリの登録削除用のロジックの追加などが発生するが、左右両方式を混在させることも可能である。

さらに、バイパスBPを記録することにより、MIN/MAXサーチにおけるバックトラックの遅延も解消することができる。MIN/MAXサーチとは、近

似候補エントリ群から近似結果を確定させるための検索である。ところで、右倒れ型であっても左倒れ型であっても、テーブル型ビットツリーの場合、照合継続中に毎段で近似候補が更新されていくわけではない。ある段で近似候補が更新され、その後、数段の近似候補更新を伴わない照合継続があり、下段のほうで照合失敗となった段階で、上段で更新された近似候補を使い、結果を確定するためにさらにMIN/MAXサーチをおこなう必要がある場合がある。

図12に一例を示してある。上位のテーブル28から下位のテーブル28に照合継続し、照合失格となった段階で、近似仮定に基づき上位のテーブル28に戻り、別ロジック(MIN/MAXサーチ)で再びツリー状にテーブル28を降りていくことになる。これはバックトラックとすることができる。照合失敗によりMIN/MAXサーチでツリーを辿るロジックが変わるので、MIN/MAXサーチでは再度のバックトラックは発生しないが、照合失敗位置と近似仮定位置が離れている場合、バックトラックと目されるMIN/MAXサーチのコストは重い。

これに対し、図13に示すように、バイパスBPを記録しておけば、一度バイパスBPの値を読むことにより、最大または最小のエントリが確定するため、実質的にはサーチプロセスとなるような時間の係る処理は不要となり、実質的にはバックトラックの発生を抑えられる。図13は、右倒れ型の例であり、テーブルA1で近似仮定更新・照合継続となり、テーブルA2では近似仮定更新無し・照合継続となり、テーブルA3で近似仮定更新無し・照合失敗となる。このとき、テーブルA3での照合失敗で近似確定のため、MAXサーチに移行するが、最新の近似仮定はテーブルA2'であるため、テーブルA2'にアクセスする。そして、テーブルA2'にバイパスBPが記録されていれば、それにより直接、最下段の最大のエントリまで移動でき、照合過程は終了する。

検索装置、分類装置およびバイナリビットツリー照合器

図14に検索装置10の概要を示してある。この検索装置10は、各種の検索を行う分類器15を複数備えており、各種の検索を並列に行えるようになっている。したがって、検索装置10には、複数の分類器15により実現される複数の

分類過程を有する分類方法が実装されている。1つの分類器15により各種の検索をシーケンシャルに行うことも可能であり、本発明の検索装置はこれに限定されるものではない。

- この検索装置10に対しては上流のアプリケーション、たとえば、図1に示した制御ユニット12、8c、7c、6cから検索対象ビットストリングであるキー21と、その検索種別を示すルートTAG31と、検索ジョブを識別する情報(ID)32とが供給される。たとえば、SPD検索機能8であれば、1つのパケットデータ ϕp を管理するために、ソースIP、デスティネーションIP、ソースポート、デスティネーションポートおよびプロトコルを示すビットストリングを少なくとも照合して、それらの全てを満足するルールを検索する必要がある。このような場合は、ソースIP、デスティネーションIP、ソースポート、デスティネーションポートおよびプロトコルを示すビットストリングをそれぞれ異なる分類装置15に対して検索対象ビットストリング(検索対象キー)21として供給し、それぞれのビットストリングと比較するビットパターン(エントリ)22が登録されたバイナリビットツリーの最上位の部分テーブル25(28)を示すメモリ11あるいは8mのアドレスをルートタグ31として供給する。これにより各々の分類装置15においては、供給された検索対象キー21を、その検索対象キー21の範囲検索に用いるべく予め登録されているビットパターン(エントリ)22と照合することができる。
- さらに、検索装置10は、複数の分類装置15に供給された検索対象キー21の照合結果により得られるルールの論理積を求めて最終結果を取得するための論理演算装置(AND化器)19を備えている。したがって、この検索装置10は、複数の分類器15により実現される複数の分類過程と、AND化器19により実現される論理演算過程とを有する分類方法が実装されている。各々の分類装置15における検索処理時間は、各々の分類装置15に供給される検索対象キー21のビット長により異なり、さらに、照合過程において上位の部分テーブル28に戻るバックトラックが発生するか否かによっても異なる。したがって、分類装置15から同期して検索結果が出力されない可能性がある。このため、分類装置15に供給される検索対象キー21にID32を付加することにより、論理演算装

置 19 における論理演算を可能とし、さらに、パケット管理装置 3 において管理されるパケットデータ ϕp との関連付けも可能なようにしている。

本例の分類装置 15 は、パイプライン状に接続された複数の照合装置 50 を備えており、これらの照合装置 50 の間ではデータパケットとして標準化された照
5 合継続情報（入出力パラメータ）33 が伝達され、パケット駆動される構成と
なっている。したがって、分類装置 15 は、複数の照合装置 50 により形成され
たデータフローを備えており、複数の照合段階を備えた分類過程を有する分類方
法が実装されている。また、それぞれの照合装置 50 が各照合段階の処理を行う
ので、複数の照合段階により多段階の照合を行う照合方法が分類装置 15 に実装
10 されているということもできる。この分類装置 15 は、クロック同期のハード
ウェアで実現することも可能であるし、データフロー型のハードウェアで実現す
ることも可能である。分類装置 15 は、さらに、圧縮された部分テーブル 28 が
記録されたメモリ 11 と照合装置 50 との間の入出力速度を向上するためのバース
ト転送バッファ 16 も備えている。

15 以下で説明する検索装置 10、分類装置 15 および照合装置 50 の構成は、ソ
フトウェアで実現することも可能であるし、ソフトウェアを主体として部分的に
ハードウェアを用いて実現することも可能である。しかしながら、実行速度を考
慮した場合、できるかぎりハードウェアで実現することが望ましい。本部分テー
ブル 25 あるいは 28 を用いた検索方法は、単一、あるいはほぼ単一の構成の照
20 合装置（照合器）50 をパイプライン状に接続することによりビット長の変化に
対応でき、また、ルートタグ 31 といった検索種別を示す情報が提供されること
により単一の構成の照合装置 50 により多種類の検索を実行できるようになっ
ている。したがって、この照合方法は、非常にハードウェア化しやすい照合方法で
あり、経済的で、処理速度の速い検索装置を提供できる。また、上位のソフト
25 ウェアあるいはアプリケーションからは、検索対象キー 21 と、検索種別情報 3
1 を提供するだけで、所望の検索を実行できるので、上位のアプリケーションの
負荷も非常に小さい。

分類器の構成

図15に、8個の4ビット型の照合器50をパイプライン状に繋いで32ビットの検索対象キー21に対する検索を行う分類装置15の概略構成を示してある。この分類装置15は、複数の照合器50をハードウェアとして備えており、それらの照合器50をソフトウェアあるいはハードウェアで接続している構成である。照合継続情報33に含まれるルートタグ (RootTAG) 31および照合タグ (Next TAG) 34は、SDRAMメモリ11の上の圧縮テーブル28を指すポインタ (アドレス) である。それぞれが照合段階を実行する各照合装置50は、照合継続情報33に含まれる検索対象キー21から順番に4ビットずつを部分対象キー36として受け取り、ルートタグ31または照合タグ34により指定されるテーブル28からマスクデータ28aを読み取り、近似一致による比較を行う。比較によりヒットしたエントリ (部分エントリ) 26に対応する次段の圧縮テーブル28のポインタ (NextTAG) 34を、圧縮テーブル28のアドレステーブル28bから読み取り、照合継続情報33に含めて次の照合装置50に供給する。

- 15 検索対象キー21や照合タグ34などの照合継続情報33は、別々に供給されても良い。しかしながら、データパケットとしてまとめて入力することにより、データフロー化しやすい。また、検索対象となる部分対象キー36は、上位のプログラムにより予め分割された状態で照合装置50に供給されても良い。しかしながら、検索対象キー21の値は全ビットが照合継続情報33に含められ、シフトしながら必要なビット数ずつを部分対象キー36として使用していくことが望ましい。これは、照合器パイプライン構成の途中でパイプラインステージを逆流 (バイパス等によるバックトラック) する可能性があり、検索対象キー21を照合器50の外部で切り分けたのでは、パイプラインの進行 (クロック) 同期で適切なステージ (段階) へ投入していくことができないためである。また、SDRAM11に対するアクセスの状態 (競合・排他) や、データキャッシュをつけた場合のオンキャッシュかどうか等で各照合装置50における処理速度は変動する可能性が高い。したがって、照合器パイプラインの進行自体をクロック同期ではなく要求応答型 (データフロー型) とした方が処理効率は高く、この点でも、検索対象キー21を照合継続情報33に含め、さらに、照合検索情報33をデータ
- 20
- 25

パケット化して次の段階の照合装置 50 に供給するメリットがある。

図 16 には、分類装置 15 の異なる例を示してある。この分類装置 15 は MAX/MIN サーチを実行する際のバイパス用のバス (BP バス) 39 を備えており、照合装置 50 は、BP バス 39 に出力するインターフェイスを備えている。

- 5 したがって、この分類装置 15 においては、各照合装置 50 が非同期でテーブル 28 にアクセスする頻度が上昇する。図 15 に示したように、各段のテーブル 28 を同一の SDRAM 11 におくと、パイプラインを構成する各照合器 50 のメモリアクセス競合があるため、パイプラインが有効に機能しない。従って、各段のテーブル 28 を個別のメモリ 11a におき、各照合器 50 のメモリアクセスを
10 独立としたほうがパイプラインの有効性が高い。この照合装置パイプラインを備えた分類装置 15 では、1 つの照合器 50 の処理に要する時間 (メモリアクセスを含む) を「T」とし、32 ビットの検索対象キー 21 が途切れなく入力されれば、検索のスループットは「T」(本当は $1/T$) となり、レイテンシは「8 T」となる。

- 15 照合器 50 の構成は、SDRAM 11 を除いて処理時間が一定するので、パイプラインを組みやすい。データキャッシュ制御により SDRAM アクセスを排除する履歴キャッシュ方式や専用 SRAM 型履歴キャッシュ方式を使えば、SRAM 11 に対する実質のアクセス時間を一定化でき、パイプラインをさらに構成しやすい。これに対し、共有型データキャッシュであれば、キャッシュのスループットと全パイプラインステージのスループットの合計が等しいか、あるいは共有型データキャッシュの方がスループットを高くする必要がある。
20

- 図 17 に、分類装置 15 のさらに異なる例を示してある。上記の例では、複数の照合器 50 を処理パイプラインとして並べた構成であるが、省リソースを優先した場合は、図 17 に示したように、1 個の照合器 50 を繰り返し使う構成も可能である。この場合、照合器 50 からのメモリアクセスはある時点では常に 1 アクセスとなるので、テーブル 28 を全て同一の SDRAM 11 に置くことで処理速度の優劣はない。
25

照合装置の構成

図 18 に照合装置 50 の構成を示してある。この照合装置 50 は、検索対象ビットストリング（検索対象キー）21 を複数ビットの部分対象ビットストリング（部分対象キー）36 に分けて、予め登録された複数の登録ビットパターン（エントリ）22 と多段階で照合するための 1 つの照合段階を実行する照合装置である。照合装置 50 は、照合継続情報 33 に含まれるコマンド 37 をデコードして照合装置 50 における処理を制御するコマンドデコードユニット 51 と、検索対象キー 21 から、現段階の部分対象キー 36 を選択し、その部分対象キー 36 が取り得る全ての値と照合する比較ユニット（全照合手段）52 と、メモリ 11 からその照合段階の圧縮テーブル 28 のマスクデータ 28a をロードするロードユニット（パターンロード手段）53 とを備えている。ロードユニット 53 は、圧縮テーブル 28 をメモリ 11 からバッファ 59 に格納するテーブルロード部 53a と、バッファ 59 に格納された圧縮テーブル 28 からマスクデータ 28a を読み出すマスク読み出し部 53b とを備えている。圧縮テーブル 28 にマスクデータ 28a と共に格納されているアドレステーブル 28b およびバイパスデータ BP は多くの処理で必要になる。このため、本例では圧縮テーブル 28 をバッファ 59 にロードしている。マスクデータ 28a は、上述したように、エントリ 22 の部分ビットパターン 26 であって、先行する段階から得られる照合継続情報 33 により決まる、部分対象キー 36 に対応する段階（現段階）の部分登録ビットパターン（部分エントリ）26 をフラグ（ビットパターン）により示すものである。

照合装置 50 は、さらに、比較ユニット 52 における全照合結果とマスクデータ 28a により、部分対象キー 36 に一致する部分エントリ 26 の有無、すなわち一致を含む照合結果を示すことができる判定ユニット 54 を備えている。また、照合装置 50 は、判定ユニット 54 の照合結果により、アドレステーブル 28b から、照合により特定された部分エントリ 26 に続く次の段階のマスクデータ 28a のアドレス（タグ（本例では圧縮テーブル 28 のアドレス））を照合タグ（照合候補）34 として出力する出力ユニット 55 を備えている。この照合装置 50 は、一致検索のみではなく範囲検索も行えるように、判定ユニット 54 は、

部分対象キー 36 に値が最も近い最大（右倒れ型近似について記載し、左倒れ型近似については、最大を最小に変更することによりほぼ対応できる）の部分エントリ 26 の有無を示すことが可能である。本例の判定ユニット 54 は、比較ユニット 52 の出力をマスクデータ 28a によりマスク処理するマスク器（VLD、
5 VaLiDation）54a と、マスク器 54a の出力から状態を選択する選択器 54b とを備えている。

出力ユニット 55 は、判定ユニット 54 により一致と判断された部分エントリ 26 に続く次の段階のマスクデータ 28a のアドレス（圧縮テーブルのアドレス）を照合タグ 34 として出力する。さらに、出力ユニット 55 は、判定ユニット 54 において、最大の部分エントリ 26 が見つかったときは、その最大の部分
10 エントリ 26 に続く次の段階のマスクデータ 28a のアドレス（圧縮テーブル 28 のアドレス）を範囲検索のための候補アドレス（候補タグ、近似候補）35 として出力し、照合継続情報 33 のパケットデータに保存する。したがって、一致する部分エントリ 26 があり照合継続の場合であっても、最大の部分エントリ 2
15 6 が見つかる範囲検索用の候補タグ 35 がパケットデータに保存される。このため、次の照合段階において、一致する部分エントリ 26 がなく、照合終了で、さらに、その照合段階では最大の部分エントリ 26 がないときに、候補タグ 35 を照合タグ 34 として照合継続情報 33 に出力することが可能となる。この照合タグ 34 により MAXサーチに移行することができる。

20 さらに、照合装置 50 の出力ユニット 54 は、MAXサーチに移行したときに、圧縮テーブル 28 からバイパスデータ BP を読み取って最終照合結果として出力することができる機能も備えている。したがって、出力ユニット 55 は、照合状態などを反映したコマンド 37 を生成するコマンド更新部 55a と、照合タグ 34 を更新する照合タグ更新部 55b と、候補タグ 35 を更新する候補タグ更新部
25 55c と、バイパス読み出し出力部 55d とを備えている。そして、出力インターフェイス 56 からは、出力ユニット 55 により更新されるパラメータ 34、35 および 37 と、入力された照合継続情報 33 に含まれる ID 32 と、検索対象キー 21 とを含むデータパケットが照合継続情報 33 として、パイプラインの下流に位置し、次の段階の照合を行う照合装置 50 に供給される。

圧縮テーブル 28 にバイパスデータ B P が設定されていないときは、照合終了を示すコマンド 37 を受け取ると、出力ユニット 55 の照合タグ更新部 55 b がマスクデータ 28 の最大の部分エントリ 26 に続く次の段階のマスクデータのアドレスを照合タグ 34 として出力し、MAXサーチが継続される照合装置 50 のパイプラインを制御する。

さらに、本例の照合装置 50 は、マスクデータ 28 a およびアドレステーブル 28 b を更新することにより、エントリを追加または削除する更新ユニット 60 と、更新されたテーブル 28 をメモリ 11 に書き込むテーブルライトユニット 61 とを備えている。更新ユニット 60 の動作については、以下でさらに詳しく説明する。

照合装置における処理

図 19 に、照合装置 50 における照合処理（照合方法）50 m をフローチャートにより示してある。この照合装置 50 は、各々の照合装置 50 が 1 つの照合段階を受け持ち、図 19 に示す照合処理 50 m は 1 つの照合段階（現段階）の処理を示している。ステップ 101 で、コマンド 37 がエントリの追加削除を示しているときは、ステップ 102 で、更新ユニット 60 によりテーブル 28 を更新する（更新工程および更新過程）。ステップ 103 で、コマンド 37 から、現段階に先行する前段階の照合が近似で照合終了していることが判明すると、以下に説明する現段階の照合のステップ 104、105 および 106 をバイパスし、出力ステップ 110 に含まれるステップ 111 でバイパス読み出し出力部 55 d によりバイパスデータ B P を図 16 に示したバイパスバス 39 に出力して照合を終了する（バイパスする工程）。

コマンド 37 より、現段階の照合を継続する場合は、ステップ 104 において、比較ユニット 52 により現段階の部分対象キー 36 を比較する（全照合工程）。ステップ 105 において、ロードユニット 53 により現段階のマスクデータ 28 a をロードし（パターンロード工程）、ステップ 106 において、判定ユニット（VLD ユニット）54 により現段階の照合結果を得る（判定工程または VLD 工程）。そして、ステップ 107 において、一致する部分エントリ 26 があれば

照合継続なので、ステップ109において候補タグ更新部55cにより候補タグ35を更新し（候補アドレス記憶工程）、出力過程110のステップ114で照合タグ更新部55bにより照合タグ34を更新し、コマンド37をコマンド更新部55aにより照合継続にセットして出力する（出力工程）。

- 5 一致する部分エントリ26がない場合は原則としては照合終了である。しかしながら、ステップ108で、近似の部分エントリ（最大の部分エントリ）26がある場合は、出力ステップ110に含まれるステップ113において、照合タグ更新部55bが近似の部分エントリ26に続くタグアドレスを照合タグ34にセットして出力する。ステップ108で、近似の部分エントリ26がない場合は、
- 10 出力ステップ110に含まれるステップ112において、照合タグ更新部55bが、上流の照合段階で得られ、照合継続情報33に含まれて伝達された候補タグ35を照合タグ34にセットしてバイパスバス39に出力する。これにより、MAXサーチが開始される。

- これらの工程を備えた照合方法50mでは、実際には、独立して処理できる工程は並列に実行される。たとえば、コマンドを解釈する工程101および102、
- 15 部分対象キー36を比較する工程104と、マスクデータをロードする工程105とは独立した処理であるので、並列に実行できる。テーブルをロードするフェーズはSDRAMアクセスであるため時間がかかり、比較フェーズなどとは並列に実行することによりメモリアccessのオーバーヘッドを緩和できる。したがって、読み出し速度が非常に速い特殊なメモリを用いなくても、本例の照合装置50においては十分な検索スピードを確保できる。また、出力ステップ110において異なるパラメータを更新するフェーズも相互独立であり、コマンド更新部55a、照合タグ更新部55bおよび候補タグ更新部55cも独立しており、
- 20 これらにおける処理は並列に実行可能である。したがって、図20に示すように、通常の照合モードの場合、照合装置50においては、テーブルロード53と比較52などが並列で実行される。

本例のロードユニット53は、照合タグ34の指示により、SDRAMメモリ11の上の圧縮テーブル28を照合器50の内部のテーブルバッファ59にロードする。圧縮テーブル28のデータは、必ずしも全て利用されるものではないが、

少なくともマスクデータ 28 a と、アドレステーブル 28 b の 1 個あるいは 2 個のアドレス (タグ) の値は参照される。したがって、各照合器 50 において共有資源である SDRAM 11 に対し、2~3 回のリードが発生するよりは、バーストリードにより 1 回だけ SDRAM 11 からリードし、照合器内部で必要値を読み出すほうが効率的である。上述したように、SDRAM からバーストリードするには時間がかかるため、このテーブルロードのフェーズからマスクデータ読み出しフェーズは、比較フェーズと並列に実行される。詳しくは後述するが、SDRAM 11 との間にデータキャッシュを置いた場合、照合器 50 に内蔵のテーブルバッファは不要であるが、その場合でも、共有資源たるデータキャッシュへのアクセスを複数回おこなうよりも、やはりローカルにコピーしてアクセスするほうが効率的な場合がありえる。

比較ユニットの構成

図 2 1 に、比較ユニット 52 をさらに詳しく示してある。この照合装置 50 においては、部分対象キー 36 のビット数が固定なので 1 テーブルのエントリ数は固定である。また、圧縮テーブルなのでエントリとビットパターンの対応も固定である。したがって、部分対象キー 36 の値と各エントリとの比較は固定論理回路により構成可能である。

この照合装置 50 は 4 ビットの比較を行う。このため、16 個の比較器 CMP 0 から CMP 15 を備えており、比較器 CMP 0 (エントリ No. 0) は入力検索対象キー 36 と値 0 との比較、比較器 CMP 1 (エントリ No. 1) は入力検索対象キー 36 と値 1 との比較、そして、比較器 CMP 15 (エントリ No. 15) は入力検索対象キー 36 と値 15 との比較を行う。このように各比較器 CMP i で、入力値 x と固有値 i との比較をおこなう。つまり、比較器 CMP i (i は 0~15) はそれぞれのエントリに固有の論理回路である。

比較ユニット 52 は、比較器の代わりにルックアップテーブル LUT を用いたハードウェアなどの他のハードウェアで構成することも可能であり、本例に限定されるものではない。

図 2 2 に示すように、各比較器 CMP i は入力値 x として対象キー 4 ビットを

取り、それぞれのCMP iは、それぞれの固有値 i との比較を行う。各CMP iは、入力値 x と固有値 i との比較結果「<」「=」「>」の3値（2ビット）をCMP i 出力として出力する。各比較器CMP iは、この比較論理に基づき、比較結果を出力するよう比較器毎に論理圧縮され、最適化された回路で構成される。

- 5 図23は、比較ユニット52の異なる例を示している。並列に実行されるテーブルロードフェーズ～マスク読み出しフェーズに時間がかかることを考えれば、必ずしも上記のように16個の比較器CMP iを並列する必要はない場合がある。図23は、1個の比較器CMP xにより比較動作を16回繰り返す構成である。他の処理との処理時間のバランスが取れるのであればハードウェアを削減できる。
- 10 もちろん、テーブルロードフェーズ～マスク読み出しフェーズの所要時間との兼ね合いで、比較ユニットに使用される比較器CMP iの必要個数は1個に限定されることはなく、2個以上であっても良い。

判定ユニットの構成

- 15 図24に、判定ユニット54のマスク器54aの構成を示してある。マスクフェーズではマスク器VLD 54aにより無効エントリのマスク処理がおこなわれる。各エントリをマスクするマスクエレメント54eは、各エントリに対し同一の論理回路である。マスクデータ28aは、実際のエントリ（部分エントリ）26の有効・無効を各1ビットのマスク値で表現したものである。マスクデータ28aにより、比較フェーズでの各エントリ（仮想的なエントリ）の比較結果から、無効エントリの比較結果を取り除き、実際に登録されている各エントリ26との比較結果に変換する。
- 20

- 各マスクエレメント54eは、図25に示すように、比較フェーズのCMP i 出力（2ビット）とマスク値（1ビット）の合計3ビットを入力とし、「×」
- 25 「<」「=」「>」の4値（2ビット）を出力とする。ここで、「×」は無効エントリ、「<」「=」「>」は有効エントリである。

図26から図28に、判定ユニット54の選択回路54bの選択論理を示してある。マスクフェーズ後の各エントリのVLD出力は、「×」（無効）、「<」（キー36<エントリ26）、「=」（キー36=エントリ26）、「>」

- (キー 3 6 = エントリ 2 6) の 4 状態である。選択フェーズでは、これらの V L D 出力の中から完全一致検索、範囲一致検索の照合結果を得て、次段へ繋がるエントリを選択する。図 2 6 に示した選択論理は、完全一致検索の選択論理であり、マスク器 V L D 出力が「0 1」となるエントリを選択する。完全一致を示す出力関数 $G_f(i)$ 、出力 $G_f(i)$ が 1 となるエントリがあったかどうかを意味する出力関数 S_f 、完全一致のエントリを示す関数 N_f は以下の通りである。

$$G_f(i) = \overline{\text{Bit 1}} * \text{Bit 0}$$

$$S_f = \sum G_f(i)$$

$$N_f = G_f^{-1}(1)$$

($G_f(i) = 1$ となる i を求める逆関数) $\dots (1)$

- 図 2 6 に示したように、V L D 出力が「0 1」のエントリのみ出力 $G_f(i)$ (完全一致を示す出力) が「1」となる。したがって、出力 $G_f(i)$ が 1 となったエントリ N_o を取得することにより、完全一致のエントリ N_f を得ることができる。バイナリビットツリー方式ではエントリとしての重複登録はありえない。従って、 $G_f(i)$ の 1 6 出力のうち、完全一致エントリがあれば、唯一であることは保証される。出力 $G_f(i)$ による「=」の選択は範囲検索においても照合継続処理で使用する。

- 図 2 7 (a) に右倒れ一致検索型の範囲検索の照合継続選択論理を示し、図 2 7 (b) および図 2 7 (c) に、中間出力と漸化式により近似候補の出力が得られる論理を示してある。また、図 2 8 (a) に左倒れ一致検索型の範囲検索の照合継続選択論理を示し、図 2 8 (b) および図 2 8 (c) に、中間出力と漸化式により近似候補の出力が得られる論理を示してある。右倒れ一致検索の場合の近似候補を示す出力 $G_r(i)$ 、中間出力 $F_r(i)$ 、漸化式 $H_r(i)$ 、近似候補のエントリ N_o を示す出力 N_r 、出力 $G_r(i)$ が「1」となるエントリの有無を意味する出力 S_r は以下の通りである。

$$F_r(i) = \text{Bit } 1 * \text{Bit } 0$$

$$H_r(-1) = 0 \text{ (漸化式仮想初期値)}$$

$$H_r(i) = H_r(i-1) + F_r(i)$$

$$G_r(i) = \overline{H_r(i-1)} * H_r(i)$$

$$5 \quad S_r = \Sigma G_r(i)$$

$$N_r = G_r^{-1}(1)$$

$$(G_r(i) = 1 \text{ となる } i \text{ を求める逆関数}) \quad \dots (2)$$

また、左倒れ一致検索の場合の近似候補を示す出力 $G_r(i)$ 、中間出力 $F_r(i)$ 、漸化式 $H_r(i)$ 、近似候補のエントリ N_r を示す出力 N_r 、出力 $G_r(i)$ が「1」となるエントリの有無を意味する出力 S_r は以下の通りである。

$$F_l(i) = \text{Bit } 1 * \overline{\text{Bit } 0}$$

$$H_l(16) = 0 \text{ (漸化式仮想初期値)}$$

$$15 \quad H_l(i) = H_l(i+1) + F_l(i)$$

$$G_l(i) = \overline{H_l(i+1)} * H_l(i)$$

$$S_l = \Sigma G_l(i)$$

$$N_l = G_l^{-1}(1)$$

$$(G_l(i) = 1 \text{ となる } i \text{ を求める逆関数}) \quad \dots (3)$$

20

範囲検索の場合、照合継続は完全一致同様の「キー36＝エントリ26」の選択であり、照合終了の選択は、範囲検索を右倒れ一致型でおこなうのであれば「キー36＞エントリ26」となる最大エントリを選択する。左倒れ一致型であれば、「キー36＜エントリ26」となる最小エントリを選択する。したがって、右倒れ一致の場合はVLD出力が「11」のエントリの中間出力 $F_r(i)$ が「1」となり、左倒れ一致の場合はVLD出力が「10」のエントリのみが出力 $F_r(i)$ が「1」となる。

25

そして、範囲検索において、近似候補を示す出力 $G_r(i)$ では、出力 $F_r(i)$ が「1」のエントリのうち、最大値のエントリ（エントリ番号が最大のエ

ントリ)のみが「1」となり、その近似候補のエントリNoを示す出力 N_r が決定する。

同様に左倒れ一致型の場合は、出力 $G_r(i)$ では、出力 $F_r(i)$ が「1」のエントリのうち、最小値のエントリ(エントリ番号が最小のエントリ)のみが「1」となり、近似候補となるエントリNoを示す出力 N_r が決定する。

図29に、右倒れ一致型の場合の出力の変化を示してある。図29(a)は比較フェーズ後のCMPi出力であり、図29(b)は、テーブルより読み出したマスクデータ28aである。図29(c)はVLD出力を示し、ビット毎の(a)×(b)となる。図29(d)は「キー値>エントリ値」となるエントリ群の抽出(中間出力) F_r であり、図29(e)の中間出力 H_r は、出力 F_r から右エッジ(「キー値>エントリ値」となる最大エントリ)を抽出したものである。図29(f)は、最終出力 G_r であり、出力 H_r を利用し、「キー値>エントリ値」となる最大エントリのみを出力する。これを見ても $G_r(i)$ が「1」となるエントリ、すなわち近似候補となるエントリは1箇所であることは明らかである。

図29(g)は一致検索の出力 G_r を示しており、VLD出力をもとに「キー値=エントリ値(01)」となる一致エントリを抽出している。この照合方法においては、テーブル28への重複登録がない構造なので、この一致エントリの唯一性は自明である。出力 G_r でも出力 G_r でも、演算の結果、該当エントリは存在しないか唯一存在のいずれかとなる。

図30(a)から図30(g)に、左倒れ一致型の場合の出力変化を示してある。右倒れ一致型の場合と同様に G_r の16出力のうち、 $G_r(i)$ が「1」となる近似候補のエントリは唯一であることは保証される。

25 出力ユニットの構成

出力ユニット55においては、判定ユニット54から得られる上記の照合結果に基づいて各パラメータを更新する。照合タグ更新部54bにおいては、選択回路54bにより得られるエントリNo(出力 N_r)に従い照合タグ34を更新する。選択回路54bより得られる出力 S_r を見て、出力 S_r が「0」であれば更

新はおこなわない。照合継続となるエントリがなく候補が確定するからである。出力 S_r が「1」であれば、照合器 50 にロード済みの現段階の圧縮テーブル 28 のアドレステーブル 28 b から、エントリ N_r に対応するタグ情報（アドレス）を読み出し、照合タグ 34 を更新する。圧縮テーブル 28 においては、

- 5 「ワード位置+0」がマスクデータ 28 a であり、「ワード位置+1」からタグ情報 28 b が記録されている。

- 候補タグ更新部 55 c においては、選択回路 54 b により得られるエントリ N_o （出力 N_r ）に従い候補タグ 35 を更新する。候補更新情報である出力 S_r を見て、出力 S_r が「0」であれば更新はおこなわない。新規候補となるエントリ
- 10 がないからである。候補更新情報 S_r が「1」であれば、照合器 50 にロード済みの現段階の圧縮テーブルから、エントリ N_r に対応するタグ情報を読み出し、候補タグ 35 を更新する。ここで候補タグ情報 35 には、タグ情報だけでなく、照合段階を示す候補段階情報も記録することが有効である。この候補段階情報は、候補タグに対応するテーブル段位置に設定する。つまり、候補タグがどのテーブル
- 15 段であるかを示す情報である。この候補段階情報は、パイプラインが照合段数でフィックスされている分類装置 15 においては、候補確定後のバイパス処理で必要となるものである。

- タグアドレスからテーブル段位置が一意に決定できるようにテーブル 28 が配置されている場合は、候補段階情報は候補タグ情報に含まれるため、別処理は不要である。そうではない場合、照合器 50 は自分がどのテーブル段数の処理を担当するのかの情報を保持し、その情報に従い、次段の情報を候補段階情報として候補タグ情報 35 の中にセットする。照合器 50 をパイプライン状に接続して利用する場合、テーブル段位置とはつまり照合器 ID であるので、自器 ID が分かり、自器に対しパイプライン次段にあたる照合器 ID が分かればよい。パイプ
- 20
- 25 インの次段の照合器との関係は固定なので、照合器 ID は、照合器の初期化時に定めればよい。次段の照合器 ID を使用するよりは、テーブルのタグアドレスに照合段階 ID を含めてしまうほうが簡素である。データキャッシュを用い、データキャッシュのコントロールをおこなう場合、テーブルは段位置別にグループ化されたアドレスとなるので、それを流用できる。

コマンド更新部 55 a は、選択回路 54 b により得られる照合継続情報である出力 S_f に従い、次段に供給するコマンド 37 を更新する。照合継続情報である出力 S_f が「0」であれば、照合継続がなくなり、候補が確定したことを意味する。したがって、「CMD=MATCHRDBP」を出力する。これにより、次段の
5 照合装置 50 では候補タグ 35 と一致している照合タグ情報 34 によるバイパス処理が行われ最終結果が出力される。照合継続情報 S_f が「1」であれば、照合継続であるので、「CMD=MATCHEXEC」を出力する。これにより、次段の照合器装置でも照合タグ 34 により継続して照合が行われる。

出力ユニット 55 は、さらに、コマンド更新部 55 a により決定した次のコマ
10 ンド 37 の値に従い、出力先を決定する。次コマンド 37 が「CMD=MATCHEXEC」であれば、照合は継続するので次段の照合装置 50 にパケット化された照合継続情報 33 を出力する。次コマンド 37 が「CMD=MATCHFIN」であれば、バイパス読み出し後の出力処理であるから、出力をスルーとして最終段の小照合器 50 の次（出力器）に送り出す。スルー出力が照合器パイプラインを順次通過
15 しながら出力されるのであれば、次の照合器 50 に最終出力を送り出す。バス 39 を使ってショートカットするのであれば、バス 39 に最終出力を出力する。

次コマンド 37 が「CMD=MATCHRDBP」であれば、照合は終了し候補が確定するので、候補タグ情報 35 によるバイパス処理をおこない、検索結果を決定する。この際、候補タグ情報 35 は必ずしも次段テーブルを指しているわけではなく、それ以前の段を指している場合がありうる。したがって、次段の照合装置 5
20 0 に出力できるとは限らず、候補段階情報により候補タグを扱える段階の照合装置 50 を特定し、照合継続情報 33 を出力する。

図 31 にコマンド 37 の一覧を示してある。なお、これは完全一致検索、1 次元範囲検索に加え、後述する登録・削除も含めたものである。

25

バイパス処理とショートカット処理

図 32 に、図 16 に示した分類装置 15 においてバックトラックのあるバイパス処理を実行している様子を示してある。この分類装置 15 においては、バイパス用のバス 39 を備えており、バス 39 に照合継続情報 33 を出力することによ

りバックトラックを発生させている。図 16 に示した分類装置 15 はメモリアクセスタイムを改善するためにメモリ分離型にしている。このため、あるタグアドレスの指すテーブル 28 は、それに対応する段階の照合装置 50 からしかアクセスできない。ところが、バイパス処理移行（候補確定）時に得られる候補タグ情報 35 は、照合器パイプライン上の次段とは無関係に、自器より前のいずれかの照合器 50 に属するものである。従って、バイパスバス 39 により候補タグ 35 に従って処理パイプラインを逆行できるようにしている。

図 33 に、図 15 に示したメモリ共用型の照合器パイプラインを備えた分類装置 15 においてバイパス処理を実行する様子を示してある。メモリ共用型の照合器パイプラインを備えた分類装置 15 の場合、全ての照合器 50 がメモリ 11 に格納された任意のテーブル 28 にアクセスできるので、バックトラック用のバイパスバスは必要ない。メモリ共用型の照合器パイプラインでは、次段の照合器 50 がバイパス処理をおこなう。したがって、最終段での次段のバイパス処理発生に対応するため、最終段の照合器 50 の次にバイパス専用のバイパス処理器 58 を設ける。F 番目の照合器 50 で照合失敗、すなわち、A～F 番目の照合器 50 のうち最後に更新された候補が確定することにより次段で候補タグバイパス処理が発生した場合、次の G 番目の照合器 50 がバイパス処理をおこなう。G 番目の照合器 50 は本来、ある段のテーブル 28 の処理パイプラインを構成しているが、メモリ共用型のため、任意の段のテーブル 28 にアクセスできる。従って、F 番目の照合器 50 で照合失敗し、候補確定が発生し、G 番目の照合器 50 にてバイパス処理をおこなう場合、テーブル B～G のどの位置の候補であっても、そのテーブルにアクセス可能である。

バイパス処理が完了し結果タグ情報（最終照合結果）が得られて以降の照合器 50 は、結果タグ情報のスルーのみを行う。最終段の G 番目の照合器 50 では、照合継続が検索成功（完全一致エントリ）を意味し、照合失敗がテーブル B～G あるいは結果タグのいずれかを指す候補タグが得られる事を意味する。G 番目の照合器 50 で照合継続により検索成功の場合、照合器 50 から出力されるタグ情報は結果タグ情報であり、それを受けた最終段バイパス処理器 58 では、検索終了としてステータスだけを更新し、バイパス処理は不要である。

一方、G番目の照合器50で照合失敗により候補確定の場合、候補タグ情報35がテーブルB～Gのいずれかを指すタグ情報である可能性がある。したがって、バイパス処理器58はバイパス処理を行い、候補タグ情報35の指すエントリからテーブル28にアクセスしてバイパス値(BP)を読み取り、検索終了ステータスにする。G番目の照合器50で候補確定であり、候補タグ情報がG番目の照合器50で更新された場合は、テーブルGに記されたタグが結果タグ情報となる。このため、更新された候補タグ情報は結果タグ情報、すなわち最終照合結果となっている。従って、バイパス処理器58では「ステータス=検索完了」にするだけで、バイパス処理は不要である。

以上まとめると、バイパス処理器58は入力の結果タグ情報であればスルーする。そうでない場合は、テーブルB～Gのいずれかを指すタグ情報が得られるので、バイパス処理器58はバイパス処理をおこなう。結果タグ情報か否かの識別は、タグアドレスの上位ビットを識別用の情報として使うか、パイプラインを流れるデータに情報を付与することで行うことができる。

図17に示した繰り返し型の分類装置15の場合は、全ての照合器が任意のテーブル28にアクセスできるので上記と同様にバイパス用のバスは必要ない。さらに、繰り返し型の場合、制御用のカウンタ値を、バイパス処理を意味する値にし、繰り返し処理の一環として自器にてバイパス処理をおこなうことができるので、バイパス処理器58も不要である。

ところで、バイパス処理にて得られるタグ情報は最終結果であり、バイパス処理以降の照合器パイプラインはスルーしてパイプライン最終段まで進み分類装置15の出力となる。また、完全一致検索での照合不一致(検索失敗)あるいは範囲検索での照合不一致または候補なし(検索失敗)の場合も同じく、検索失敗のステータスを持って自器以降の照合器パイプラインをスルーしていく。バイパス処理のためのバックトラックがなければ、分類装置15において、検索対象キー21を入力した順番と、その照合結果が出力される順番とが保障されるが、バイパス処理のためのバックトラックを許容するので、入力の順番と出力の順番が同一であることは保障されない。このため、検索対象キー21にID32をセットにして供給するようにしている。したがって、スルーする場合も、図34に示

すように、照合器パイプラインをスルーで進んでいく必要はなく、図 3 5 に示すように、バイパス用バス 3 9 を転用して照合器パイプラインをショートカットすることが可能となる。

さらには、ショートカット先は常に最終出力部なので、図 3 6 に示すように、
5 バイパス用バス 3 9 とは別にショートカット用のバス 3 8 を設けてもよい。バイパス発生とほぼ同数ショートカットは発生するため、常に決まった宛先のためにバイパス用バス負荷を高める必要はない。バイパス用バス 3 9 のバイパス処理での利用率が 5 0 % 以下であるなら、ショートカットもバイパス用バス 3 9 を使う方が、ハードウェアが増大しないというメリットがある。

- 10 メモリ共用型照合器パイプラインを備えた分類装置 1 5 においては、バイパス用バスは不要である。このため、図 3 7 に示すように、ショートカット用のバス 3 8 を新規に設けることが望ましい。繰り返し型の分類装置の場合は、制御用カウンタ値の判定でバイパス処理後は繰り返し終了としておけばよく、ショートカット経路は不要である。

15

検索テーブルの管理（更新処理）

以下では、照合装置 5 0 のテーブル更新ユニット 6 0 においてエントリ 2 2 を追加あるいは削除する処理（更新工程あるいは更新過程）について説明する。圧縮テーブル 2 8 はマスクデータ 2 8 a とアドレステーブル 2 8 b で構成される。

- 20 したがって、マスクデータ 2 8 a にビットあるいはフラグで示された部分エントリを追加したり削除したりするだけで基本的には検索テーブル 2 8 の更新が可能であり、ソーティングや正規化などの複雑な処理は必要ない。このため、検索テーブルの管理は非常に簡略化され、高速に処理される。したがって、エントリの登録や削除といったテーブル 2 8 の管理を、検索対象キー 2 1 の照合処理と並
25 列実行させることができる。つまり、本例の分類装置 1 5 においては、削除（登録）と検索が、照合装置 5 0 がパイプライン状に接続された照合器パイプラインで並列に実行させることが可能となる。ただし、エントリを追加することにより検索テーブルに用意されるバイパス値 B P を更新する必要があるケースがあり、その判断に若干の処理が必要になる。新しい検索テーブルを生成するための処理

も必要になる。しかしながら、バイパス値BPは、検索テーブル28にいったん設定すると、照合処理において、近似候補が決定された後の処理をショートカットできる。このため、更新に時間を要しても、検索テーブルの更新時にバイパス値BPを設定しておくことには意味がある。

- 5 まず、テーブル28の管理処理と検索処理が照合器パイプラインにおいて並列に実行される状況を想定すると、照合中の整合性を保つための排他管理が要求される。検索（照合）処理においても、処理対象のテーブル28を占有すれば整合性保持は容易になる。しかしながら、ツリー上位テーブルにおけるシリアル化がボトルネックとなり、照合処理の並列実行を阻害しかねない。したがって、検索
- 10 処理においては「排他を確認するが、テーブルは占有しない」アクセスにより検索処理の間の並列性を生かす。それと共に、登録・削除処理の側、すなわちテーブル更新処理では「排他を確認し、さらに、テーブルを占有する」アクセスで整合性も維持する。なお、検索処理と登録（削除）処理の間では、先行する登録（削除）処理の占有に対し後続の検索処理が排他確認の待ちをおこなうことでシ
- 15 リアル化される。

- 検索処理は上記のように「排他を確認するがテーブルは占有しない」状態で並列に実行される。このため、登録・削除処理との関係では、検索処理は、先行する登録・削除処理に対しては排他管理により追い越しが発生しないようにする。後続する登録・削除処理については、検索処理の上位構造における並列性起因の
- 20 問題であり、下位構造たる検索処理で対応できるものではないため対処しない。

- 登録処理は「排他を確認し、テーブルを占有する」状態でシリアル化されテーブル単位で逐次実行される。また、各テーブルのバイパス情報を更新する必要があるため、その情報は下位から上位へと伝播するものであるため、登録処理の時に何らかの形で下位から上位への伝播をおこなわねばならない。考えられる方式として
- 25 は、上位で待機し下位からフィードバックする待ち合わせ方式と、上位から下位に下り、双方向リンクにより下位から上位に戻る方式との2種ある。待ち合わせ方式の場合、何らかの待ち合わせ機構をハードウェア資源で実現しなければならず、その資源量により並列数やテーブル段数等が制約を受けるが、処理速度的には有利である。ハードウェアによる待ち合わせ機構ではなくメモリによる待ち

合わせにすることも可能であるが、メモリアクセスが増えるため、処理速度面での有利性がなくなる可能性がある。双方向リンク方式の場合、資源量起因の制約はないものの、1テーブルのメモリ量増加やリンク読み出しによるメモリアクセス増加（処理速度に影響）など、メモリ周りに不利な点をかかえる。

5

待ち合わせ方式による登録処理

図38に、待ち合わせ方式による登録処理の概要を示してある。この方式では、テーブル化されたバイナリビットツリーを上から下に辿りながら更新する。その際、バイパスBPの更新が確定したテーブル28は「待ちなし（Wなし）」で更新処理をおこない、バイパスBPが未確定のテーブル28は各段で「待機中（Wあり）」にして待たせておく。新規のテーブル28nを作成する必要がある段階に到達した場合は、新規のテーブル28nを作成した後に、その段階以下の新規なテーブル28nとの連結と、上位で待機中（Wあり）（バイパスBPの確定待ち）となっているテーブル28へのフィードバックを並列におこなう。

図38において、テーブルAでは追加するエン트리22の部分エン트리26は既存（エントリが既存）であり、追加（登録）されるエントリの部分エン트리26がMAXエン트리（最大のエン트리）ではないとする。この場合、下位のテーブルに対するエントリの登録がどうなっても、このテーブルAでのバイパスBPの値に影響を与えずバイパス値BPを更新する必要はない。従って、「待ちなし（Wなし）」でBP更新も不要である。

テーブルBに追加するエン트리22の部分エン트리26は、既存（エントリが既存）であるが、その部分エン트리26はテーブルBのMAXエントリに該当するとする。この場合、下位での結果次第でバイパス値BPを更新する必要がある。このため、下位が確定するまでは、このテーブルBのバイパス値BPの更新が必要かどうか確定しないため、下位の更新待ちで「待機する（Wあり）」になる。

テーブルCでは、追加するエン트리22の部分エン트리26が新規（エントリが新規）であり、その新規エン트리26がMAXエン트리とする。この場合、このテーブルではバイパス値BPの更新が必要になる。この段階で、バイパス値B

Pの更新が発生したので、上位のテーブル28へ通知する。「Wあり」状態となっているテーブルBでは下位からの通知を受けてバイパス値BPを更新し、更に上位の待ちがあればそこへ通知する。この例では、テーブルAは「Wなし」なので上位への通知は不要である。

- 5 テーブルCから下への流れは上記の上位フィードバックと並列に進む。テーブルCで次段が新規テーブル28nであると判明するので、空テーブル管理キュー69に要求し、応答としてテーブルDになるべきテーブルのタグアドレスを受け取る。そして、テーブルCのマスクデータ28aに新しい部分エントリ26のフラグを追加し、アドレステーブル28bに新しいエントリタグを書き込む。テーブルD以降のテーブルは常に新規なので、空テーブル管理キュー69に対して新たなテーブルのアドレスを要求し、マスクデータ28aおよびアドレステーブル28bにデータを書き込む。同時に、バイパス値BPも更新する。

- 図39に分類装置15における照合器パイプラインにおいて、待ち合わせ方式でエントリ22を登録する構成を示してある。エントリ22の登録は上位から下位へ照合器パイプラインを進みながら、テーブルを辿っていく。バイパス値BPの更新保留となったら、各照合装置50に対応する待機管理ユニット68に必要な情報を記録したパケットを投入し、次段以降に進む。次段へは、直接の上位段となる親テーブルでの「待機有無」の情報が伝えられる。バイパス値BPが確定した照合装置50においては、親テーブルで「待機有り」なら、上位フィードバック(FB)を発行する。この上位FBと並列に、下位へのエントリ登録のパイプライン処理はおこなわれていく。上位FBは待機管理ユニット68で待ち合わせたパケットと合流し、保留していたバイパス値BPを更新する。保留解除によるバイパス値BPを更新する処理とは並列に、必要であれば待機管理ユニット68から更に上位保留へのフィードバックを行う。

25

照合装置においてエントリを登録する処理

図40に、待ち合わせ方式でエントリを登録する処理において、更新処理ユニット60が動作したときの照合装置50の制御的な構成を示してある。「DTAG」は結果タグ(登録データTAG)を示し、「TTAG」はテーブルタグを

示し、「PWF」は直接の上位段（親テーブル）での待機有無を示すフラグを示す。コマンド「ADD」は上位から下位への既存のエントリのタグを辿って伝播される。コマンド「ADDNEW」は途中段以降の新規テーブルを確保する。コマンド「ADDUPDT」は上位へのフィードでBPの更新を再開する。コマンド「ADDWCLR」は上位で待機している照合装置50に対するバイパス値BPが不変の通知であり、待機解除に繋がる。

照合継続情報33と同様の経路で更新用のデータ packets が照合装置50に伝達される。そして、ロードユニット53により照合タグ34で指示されたアドレスのテーブル28がテーブルバッファ59にロードされる（ステップ120）。
10 コマンドデコーダ51はコマンド37をチェックし（ステップ121）、コマンド37は追加「ADD」から開始される。まず、ステップ122で、マスクを読み出し、ステップ123で、キー値とマスク値により次段が新規かどうかを判断する。次段が新規であれば、ステップ124で、空きテーブルキュー69に対してテーブル28nを要求し、応答を待つ間に、ステップ125で、マスクデータ
15 28aの更新処理をおこなう。マスクデータ28aの読み出し値と、追加する部分エントリ26の値の論理和（OR）を演算し、追加エントリに該当するビット（フラグ）を1にする。

テーブルキュー69から空テーブルを確保した応答が得られると、ステップ126でその空テーブル28nのアドレスをアドレステーブル28bに書き込み、
20 追加処理を伝播するための出力TTAGに空テーブル28nのアドレスをセットする。さらに、ステップ127で、出力するコマンドCMD37を「ADDNEW」にセットする。

コマンド37が「ADD」であり、次段が既存であった場合、ステップ128で自テーブル28のアドレステーブル28bから次段のテーブルアドレスを読み出し、出力TTAGにセットする。ステップ129で出力するコマンドCMD37は入力コマンドと同じく「ADD」のままである。
25

入力コマンド37が「ADDNEW」であった場合、その段のテーブル28nは前段処理により確保された新規テーブルなので、ステップ130でテーブル初期化をおこなう。これはマスクデータ28aだけの処理でよく、マスク値として該

当するエントリビットを1に、他を0にする。初期化以降は、コマンド37が「ADD」で新規エントリであった場合と共通の処理である。エントリの登録を照合装置パイプラインに伝播する過程で、一度、コマンド37が「ADDNEW」、すなわち新規エントリになれば、それ以降の照合装置50に供給されるコマンド37は常に「ADDNEW」であり、上記の処理が繰り返される。また、上位のテーブル28にフィードバックされるバイパス値BPを更新する処理については、コマンド37が「ADDNEW」の場合は常におこなう。

- コマンド37が「ADD」の場合、すなわち、現段階のテーブル28は既存のテーブルである場合は、ステップ131で、エントリの位置によって判断する。
- 10 コマンド37が「ADD」で、該当（新規）のエントリ26が新規（次段TBLが新規）のエントリ22の部分エントリで、それが最大エントリであった場合（MAXエントリ更新）、バイパス値BPは更新が確定する。コマンド37が「ADD」で該当（新規）のエントリ26が新規のエントリ22の部分エントリで、最大エントリではない場合、バイパス値BPは不変で確定する。コマンド37が
- 15 「ADD」で該当エントリ26が既存で最大エントリではない場合、バイパス値BPは不変で確定する。バイパス値BPが確定する場合、更新するのであればステップ132で更新し、ステップ133で出力PFWを「0」にセットする。一方、コマンド37が「ADD」で該当エントリ26が既存で最大エントリの場合、
- 20 下位の状況によりバイパス値BPの更新が必要かどうか確定するため、待機管理ユニット68へパケットを登録し、バイパス値BP更新を保留する。更新を保留した場合、ステップ134で、出力PWFを「1」とする。

- 更新不要、更新確定の場合、状態が定まったので、入力PWFを参照し、上位待機があるかどうかを判断する。上位待機があれば、1段上の待機管理ユニット68へフィードバック出力する。更新確定なら、ステップ135でコマンド「ADDUPDT」、更新不要なら、ステップ136でコマンド「ADDWCLR」とし、上位の更新要不要を指示する。
- 25

図41に待機管理ユニット68の動作をさらに詳しく示してある。待機管理ユニット68は、待ち合わせ処理と、その後のバイパス値BPの更新再開および上位フィードの並列化をおこなう。コマンドを見て「ADDUPDT」なら下位の段で

- バイパス値B Pが更新されたので、ステップ137で、バイパス値B Pの更新を再開する。そのために照合器50のキューにパケットが投入される。コマンドが「ADDWCLR」なら下位でバイパス値B Pの更新がなかったので、キューのパケットは破棄される。ただし、次の上位にフィードする処理は行う。また、コマンドによる処理とはまったく独立に、出力PWF（上位で待ちがあるかどうか）をみて、出力PWFが「1」なら上位へフィードする。そのとき、IDおよびCMDは下位から来たものをそのまま上位へフィードする。

- 照合装置50では、コマンド37が「ADDUPDT」なら待機していたバイパス値B Pの更新処理を実行する。上位へのフィードは待機管理側がおこなうので、バイパス値B Pを更新した後は照合器50ではパケットを破棄する。

双方向リンク方式による登録処理

- 図42に、双方向リンク方式による登録処理の概要を示してある。この方式でのバイパスへの対応は、上から下に辿りながらエントリ登録をおこない、その際にバイパス値B Pの更新が確定したテーブルでは更新処理をおこなう。バイパス値B Pの更新が未確定テーブルはそのまま下位へ移動し、待ち合わせ処理のような「待ち」は設けない。新規テーブル作成段に到達後、その段以下の新規テーブルの連結と、双方向リンクを辿りながらの上位のテーブルに対してバイパス値B Pの確定待ちへのフィードバックを並列におこなう。

- 図42において、テーブルAでは、エントリが既存であり、MAXエントリではないため、下位がどうなっても、このテーブルAでのバイパス値B Pの更新はない。テーブルBでは、エントリ既存であり、MAXエントリに該当する。従って、下位での結果次第でB P更新がありえる。下位が確定するまではこのテーブルBのバイパス値B Pの更新が必要かどうか確定しないため保留する。テーブルCでは、エントリは新規であり、その新規エントリがMAXエントリとなる。従って、このテーブルCでバイパス値B Pの更新が必要となる。バイパス値B Pの更新が発生したので上位へ通知する（双方向リストを利用し上位テーブルBへ遡る）。遡ってきたテーブルBでは、下位からの通知を受けてバイパス値B Pを更新し、更に上位へ遡っていく。バイパス値B Pの更新の必要がなくなった段で

廻りは打ち切りである。一方、テーブルCから下への流れは、待ち合わせ方式による登録と同様に処理待機上位への廻りとは並列に進んでいく。

- 図43に分類装置15における照合器パイプラインにおいて、双方リンク方式でエントリ22を登録する構成を示してある。双方リンクを辿って上位にフィードバックするので待機管理ユニットは設けられていない。上位フィードバック(FB)は、この図では専用経路であるが、バス効率に影響がなければ検索バイパス用バス39を共用してもよい。双方向リンクによる上位廻りは検索時のバイパス処理(任意の上位へ接続)とは異なり、常に直接繋がる上位テーブル(パイプライン前段の照合器50)へのフィードバックなので、ここでは専用経路による接続とした。エントリ22の登録は上位から下位へ照合器パイプラインを進みながら、テーブルを辿っていく。バイパス値BPの更新が確定した照合器50から双方リンクにより上位フィードバック(FB)を発行する。この際、待ち合わせ方式と同様に、上位FBとは並列に、下位テーブルへの辿り(新規テーブルの接続)はおこなっていく。
- この双方向リンク方式においても、並列処理は下位への流れとバイパス値BP確定後の上位へのフィードバックの2並列である。また、上位フィードバックは完全な並列ではないがある程度、パイプライン並列化できる。フィードバックによるバイパス値BPを更新する時に、すぐに上位テーブルリンクを読み出して、バイパス値BPを更新する処理とは分岐して更に上位へのフィードバックを発行することができる。ただし、検索・検索バイパスとの関係で待ち行列数が増えるため、全体のパフォーマンスを検討した上で、上位フィードバックのパイプライン並列化の実装は検討すべきである。

照合装置においてエントリを登録する処理

- 図44に、双方向リンク方式でエントリを登録する処理において、更新処理ユニット60が動作したときの照合装置50の制御的な構成を示してある。ステップ120から132の処理は、待ち合わせ方式と共通するので説明を省略する。バイパス値BPの更新をおこなうのは、コマンド37が「ADDNEW」の場合と、コマンド37が「ADD」で新規エントリがMAXエントリの場合である。前者

の場合は上位へのフィードバックは不要である。コマンド37が「ADD」から「ADDNEW」と切り替わる段のテーブルでフィードバック済のためである。したがって、ステップ140でコマンドをチェックし、後者の場合は、ステップ141で上位リンクを辿り、ステップ142で上位テーブルへ「ADDUPDTコマンドパケット」によるフィードバックを出力する。また、「ADDUPDTパケット」が得られると、ステップ143でマスクデータを読み込み、該当エントリがMAXエントリかどうかでバイパス値BPの更新を判断する。上位から下位への処理でエントリは登録済であり「ADDUPDT」を受け取った時点では常に既存エントリである。バイパス値BPの更新をおこなった場合、更に上位テーブルへのフィードバックをおこなう。最上位テーブルでは上位リンクがNULLであり、その場合、フィードバックは終了する。

待ち合わせ方式による削除処理

削除処理も更新処理であり、「排他を確認し、テーブルを占有する」状態でシリアル化されたツリーを構成するテーブルを、テーブル単位で逐次実行される処理である。検索処理との関係も上記と同様である。削除処理の場合、登録同様に下位から上位へ伝播するバイパス値BPの更新があり、この上位への遡りを登録同様に待ち合わせ管理機構でおこなうのが待ち合わせ方式による削除処理である。登録処理とは異なり削除処理の場合は上位から下位へのエントリ検索時点ではバイパス値BPの更新を確定できるケースはない。下位テーブルでエントリが0個になった場合にそのテーブルを開放し、上位へ通知し、上位テーブルのエントリを削除し、必要であればバイパス値BPを更新する処理の流れとなる。この処理は下位から上位へ伝播していく処理である。登録と比べるとバイパス値BPの更新は同じであるが、テーブル削除の上位伝播処理があるため、削除のほうが複雑になる。

図45に示すように、待ち合わせ方式では、コマンド37が削除「DEL」のときで上位から下位へ辿りながら各段で待機し、最下位から上位に向かってコマンド37を「DELUPDT」としてバイパス値BPを更新し、テーブルを削除した情報をフィードしていく。この基本的な流れは待ち合わせ方式の登録処理とほぼ同

様である。削除処理の場合、下位でテーブルが空になり開放されると、上位での
エントリ削除とそれに伴うバイパス値BPの更新が発生する。たとえば、テーブ
ルが空になると最終の照合装置50がパイプラインから切り離される。テーブル
が空ではない場合は、下位でのバイパス値BPの更新情報を上位へ反映するだけ
5 でよい。テーブルが空になった場合、下位でのテーブル削除が上位でのエントリ
削除につながり、上位でのエントリ削除の結果、バイパス値BPの更新のため新
しいバイパス値BPを与えるエントリの下位への問い合わせするバイパス値問
合わせが発生する。自テーブル内の情報では新しいバイパス値BPを与えるエン
トリのバイパス値BPは得られないからである。このコマンドが「DELUPDT」
10 のときの下位へのバイパス値問い合わせ（BP問い合わせ）は、登録処理にはな
く、削除処理に特有の処理である。

下位へのBP問い合わせは以下の流れで行われる。ステップ151でエントリ
削除にともなうバイパス値BPの更新の発生した段階で、下位へコマンド「DEL
RDBP」で問い合わせ、応答待ちの packets を待機する。下位の照合器50では、
15 コマンド「DELRDBP」を受けると自テーブルのバイパス値BPを読み出し、ス
テップ152で上位へコマンド「DELWRBP」を送ってフィードバックする。応
答待ちとの間で発火し、コマンド「DELWRBP」の処理として問い合わせで得ら
れた新しいバイパス値BPを書き込む。さらにステップ153で、バイパス値BP
の更新情報を上位へコマンド「DELUPDT」でフィードバックする。

20 図46に、待ち合わせ方式でエントリを削除（コマンドが「DEL」）する処
理において、更新処理ユニット60が動作したときの照合装置50の制御的な構
成を示してある。まず、ステップ160で、入力パラメータTAGで示される
テーブルをロードし、マスクデータ28aを読み込む。ステップ161で、マス
クデータ28aと削除対象のエントリ（部分エントリ）をチェックし、対象のエ
ントリが未登録ならステップ162でコマンド37を「DELERR」として削除結
25 果を最終出力する。

ステップ163で次段のタグ情報を読み取り、ステップ164で自テーブルが
最下段であるか否かを判断する。自テーブルが最下段ではなかった場合は、ス
テップ165で、下位からのフィードバック待ち合わせ用の packets （ID, W

ID=1, TTAG (自テーブルTAG), KEY (自テーブルエントリKEY)を含む)を待機管理ユニット68に投入する。ステップ16.6で、次段の照合器50へコマンド「DEL」(ID, TTAG=NextTAG, KEY=SHIF Tを含む)を出力する。

- 5 一方、自テーブルが最下段であった場合は、ステップ167で、エントリに対応するNextタグ、すなわち、結果タグを読み出し、削除結果として最終出力する。バイパス値BPの更新も含めた削除完了前に応答が戻るのがまずいようであれば、結果タグは上位フィードバックに載せてフローさせ、最上位まで戻りきったところで最終出力する。次に、ステップ168でエントリを削除する。
- 10 テーブル28のマスクデータ28aのエントリを削除するだけでよい。エントリを削除した後のマスクデータ28aの積算(和集合)SUM-MASKが「0」ならばフラグCEFを1(テーブルが空エントリ)にセットし、SUM-MASKが「1」ならばフラグCEFを「0」(テーブルにまだエントリが残っている)にセットする。また、エントリ削除前のマスクデータ28aと、削除後のマ
- 15 スクデータ28aを比較し、フラグNBPを設定する。削除したエントリがバイパス値BPを与えるエントリであれば、フラグNBPを「1」すなわちBP更新有りにセットする。

- ステップ169でフラグCEFが「1」とであると判断すると、自テーブルが空エントリになるので、ステップ170で自テーブルを空きテーブル管理キュー6
- 20 9に返却し、パケットのフラグNBPとDTAGをDC(ドントケア)にセットする。

- 一方、フラグCEFが0の場合、ステップ171でフラグNBPをチェックし、
- ステップ172で、残ったエントリの中から新しいバイパス値BPを与えるエントリ(MAXエントリ)を選択し、ステップ173で、バイパス値BPを新しい
- 25 BPエントリ対応するタグ情報(最終結果情報(結果タグ))に更新する。そして、パケットのフラグNBP(バイパス有効)を「1」にセットし、DTAGにバイパス値BP(新バイパスTAG値)をセットする。なお、ステップ171で自エントリがBPエントリでなかった場合、バイパス値BPの更新処理は不要である。そして、ステップ174で、コマンド「DELUPDT」により上位フィード

バックの発火用パケットを出力する。

図47に、コマンド37が「DELUPDT」のときの照合装置50における更新ユニット60の処理を示してある。この処理は、コマンド「DEL」が上位から下位に流れたときに待ちに入った照合装置50が、最下段からのコマンド「DELUPDT」を受けたときに、それをトリガとして発火する。まず、ステップ180でテーブルをロードする。ステップ181で、下位からフィードバックされたフラグCEFが0ならば下位テーブルは消えていないので、自テーブルのエントリ削除は不要である。フラグCEFが「0」のときは、ステップ182で、下位からのフラグNBPをチェックし、下位テーブルのバイパス値BPが変更になっているときは、さらにステップ183で自テーブルのマスクデータ28aと削除対象のエントリに従い、自己のフラグNBPをセットする。そして、ステップ185において、自NBPが「1」、すなわち、自テーブルのエントリがBPエントリ（MAXエントリ）であれば、ステップ186でバイパス値BPを更新する。下位のフラグNBPが「0」であれば、ステップ184で自NBPも「0」にセットしてバイパス値BPの更新は行わない。そして、ステップ187で自段が最上位であるかチェックし、最上位でなければ、ステップ188で上位の待ちへフィードバックする。

一方、下位からのフィードバックされたフラグCEFが「1」ならば下位テーブルが消えたので、ステップ189で、自テーブルのエントリ削除をおこなう。フラグCEFが「1」のとき、下位テーブルが消えたのでNBPおよびDTAGはDCである。自テーブルのエントリ削除にともなうバイパス値BPを更新する。さらに、エントリ削除後に自テーブルが空（SUM-MASK=0）ならば、ステップ190でフラグCEFOutを「1」にセットし、ステップ191で自テーブルを開放し、ステップ192で最上位であるか確認後、ステップ193で上位の待ちへフィードバックする。ただし、自テーブルが最上位テーブル（ルートテーブル）である場合、テーブル削除はおこなわない。ルートテーブルは空でも削除しない。また、ルートに対して上位の待ちは存在しないためフィードバックは行われない。

エントリ削除後に自テーブルが空ではなかった場合、すなわちフラグCEFO

ut が「0」のときは、ステップ194で、新BPエン트리 (MAXエン트리) を選択し、ステップ195および196で新BPエントリに対応する下位テーブルからのバイパス値BPの読み出しをおこなう。これは照合器パイプラインの都合により、次小照合器50によるバイパス値BP読み出しを待ち合わせ管理機構
5 で待つ方式でおこなわれる。待機側は、待ち合わせパケット (ID, WID=2, TAG, KEY) を投入し、下位への依頼側は、コマンド「DELRDBP」により発行する。MAXエントリでない場合は、ステップ197でルートであるか否かを確認し、ステップ198で上位へフィードバックする。

図48にコマンド「DELRDBP」と「DELWRBP」のときの処理を示してある。
10 上位に対するコマンド「DELUPDT」からの指示で下位にコマンド「DELRBP」が送られてくる。コマンドが「DELRDBP」であれば、ステップ201でバイパス値BPを読み出し、ステップ202でDTAGにセットする。そして、上位はバイパス値BP読み出しを待っているので、上位の待機管理68に対して待ち合わせ発火用パケット (ID, CMD=DELWRBP, DTAG=BP, WID=2) を投入する。
15

自器のコマンド「DELUPDT」からの待機パケットと下位からのコマンド「DELRDBP」完了後のパケットにより発火するコマンド指示が「DELWRBP」である。下位からのDTAGにバイパス値BPが入っているので、ステップ203では、自テーブルを表すTAGに従いDTAG値を書き込む。自テーブルにおいてバイパス値BPが確定したので、ステップ205で、コマンド「DELUPDT」による上位フィードバックを再開する。ただし、コマンド「DELUPDT」と同様に、
20 ステップ204でルートテーブルであるか否かをチェックし、ルートの場合は上位がないため、上位フィードバックはおこなわない。

25 双方向リンク方式による削除

この方式では、まず、コマンド「DEL」により上位から下位にエントリを完全一致型の検索で辿り、最下位まで降りていく点は変わらない。コマンド「DELUPDT」による最下位から上位へのバイパス値BPの更新、テーブル削除情報のフィードは双方向リストによるフィードでおこない、待ち合わせ方式のような待

機機構は用いない。したがって、図 4 5 に示した待ち合わせ方式とほぼ同様のフローになるが、上位に対するコマンドは待機管理ユニットではなく照合装置 5 0 に供給される。

- また、下位でのテーブル削除にともなう上位での B P 更新については、待ち合
- 5 わせ方式の場合同様にコマンド「DELRDBP」および「DELWRBP」でおこなうが、この処理も待ち合わせではなく双方向リストによるフロー移動でおこなう。したがって、待ち合わせ方式であれば、自テーブルの K E Y 値（削除対象のエントリ）は待機パケットが保持していたため下位からのフィードバックは K E Y を気にしなかったが、双方向リスト方式ではコマンド「DELUPDT」による上位
- 10 フィードごとに該当段の K E Y を再び使用できなければならない。従って、上位から下位へのコマンド「DEL」による移動で K E Y を消費してしまう（消してしまう）のではなく、K E Y 値は S H I F T による参照位置移動とし、コマンド「DELUPDT」のときには逆方向 S H I F T で該当段 K E Y 値を得られるようにしておく。つまり、K E Y 値は 4 ビット 8 段テーブルであれば、3 2（4 × 8）
- 15 ビットの K E Y フィールドを持ち、コマンド「DEL」では上位ビットから下位ビットへの参照位置移動、コマンド「DELUPDT」では下位ビットから上位ビットへの参照位置移動をおこなう。

- したがって、コマンド「DELUPDT」のときは、双方向リストの上位リンクにより上位テーブルタグは得られる。前述のように K E Y は参照位置の移動により、
- 20 常に該当段のキー値をアクセスできる構造である。コマンド「DELUPDT」のときの下位へのバイパス値 B P の問い合わせの際も同様に、上位リンクによる上位テーブルタグの取得、K E Y 値の適切な参照位置移動がおこなえる。したがって、下位へのバイパス値の問い合わせは、双方向リンクを用いて待ち合わせ方式とほぼ同様に行われる。

- 25 図 4 9 に、双方向リンク方式でエントリを削除（コマンドが「D E L」）する処理において、更新処理ユニット 6 0 が動作したときの照合装置 5 0 の制御的な構成を示してある。待機管理ユニットを用いない方式なので、図 4 6 に示した制御系統図からステップ 1 6 5 が除かれた以外はほぼ同じ制御で処理は行われる。ステップ 1 7 4 で、コマンド「DELUPDT」で上位フィードパケットを出力する

際、送出先のタグアドレス $TTAG$ は上位リンクにより上位テーブルを指定し、エントリ KEY は上位方向へ逆シフトし、テーブル段位置と整合させる。

図 50 に、コマンド 37 が「DELUPDT」のときの照合装置 50 における更新ユニット 60 の双方向リンク方式の処理を示してある。この処理においても、待機管理ユニットを用いない点を除けば、図 47 と同じ制御で処理は行われる。なお、フィードバックする際に、送出先のタグ情報 $TTAG$ は自テーブルの上位リンクにより上位テーブルを指定、エントリ KEY は上位方向へ逆シフトしてテーブル段位置と整合させることは上記と同様である。

図 51 にコマンド「DELRDBP」と「DELWRBP」のときの双方向リンクによる処理を示してある。これらの処理においても、待機管理ユニットを用いない点を除けば、図 48 と同じ制御で処理は行われる。

上記のように、本発明においては、エントリの追加および削除に伴うテーブルの管理にソーティングや正規化などの処理が不要となるので、エントリの追加削除に伴う照合処理の遅延は従来に比べて非常に小さなものになる。範囲検索におけるエントリとルールとの関係については、右倒れ近似および左倒れ近似の説明において詳述している。たとえば、既存エントリが $[a_i, \text{無限大}] [b_{i+1}, \text{無限大}]$ $\{i \text{ は複数}\}$ の時、新エントリ $[a', \text{無限大}] [b' + 1, \text{無限大}]$ を加えるとき（登録する前に）、 $[a', \text{無限大}]$ のルールには a' でヒットするエントリの（複数の）ルールを加え、 $[b' + 1, \text{無限大}]$ のルールには $b' + 1$ でヒットするエントリの（複数の）ルールを加える。既存エントリが $[a_i, \text{無限大}] [b_{i+1}, \text{無限大}]$ $\{i \text{ は複数}\}$ の時、 $[a_j, \text{無限大}] [b_j + 1, \text{無限大}]$ を削除するとき（削除した後で） $[a_j, b_j]$ に含まれるエントリから該当ルールを除く。

25 データキャッシュ付き分類装置

図 52 に、データキャッシュ 17 を備えた分類装置 15 の概略構成を示してある。データキャッシュ 17 は、照合器パイプライン 500 を構成する各照合装置 50 とメモリである SDRAM 11 の間に設置される。図 53 に示すように、本例のデータキャッシュ 17 は、N ウェイ・セット・アソシアティブ・キャッシュ

であり（Nウェイは2ウェイや4ウェイが一般的である）、キャッシュラインのサイズがSDRAM11のバースト転送サイズ、すなわち、照合テーブル28のサイズと一致するように構成されている。データキャッシュ17のキャッシュライン17aを1つのテーブルサイズとすることで、分類装置15におけるテーブルアクセスが効率的に行われる。

また、データキャッシュ内にライン単位のメモリ排他管理機構を導入することで、キャッシュメモリ上あるいはSDRAM上のテーブルデータの排他利用が可能となる。そのためには、キャッシュのラインサイズ、ライン数、セット数を適切に構成し、その構成に合わせてテーブルを割り当てれば良く、データキャッシュのヒット率を高めることができる。そして、キャッシュシステム17を採用することにより、汎用のSDRAM11を用いながらさらに高速な分類が可能となる。

図54に、データキャッシュ17を検索種別および照合段階で排他的に利用する例を示してある。本例のキャッシュシステムにおけるテーブル28のオンキャッシュ率を高めるために、照合テーブル28の各段（たとえば、テーブルA-1とA-2）を異なるキャッシュエントリに割り当てる。また異種の照合（たとえば、テーブルA群とテーブルB群）に対してもそれぞれ独立したキャッシュエントリを割り当てる。検索種別は、上流の制御ユニットあるいはアプリケーションから与えられるルートアドレス31により簡単に識別できる。また、照合段階は、パイプライン段数に相当するので、識別は容易である。そして、照合テーブルの新規登録においても、テーブルの確保が必要になった場合、テーブル段位置に対応するキャッシュエントリにマッピングされるSDRAM11のアドレスを持つ空きテーブルを割り当てていく。

また、照合テーブル28はツリー型テーブルの構造となるので、下位段でテーブル数が広がっていく。したがって、下位段テーブルには複数キャッシュエントリを割り当てておくことが望ましい。図54においては、Aタイプの照合用のテーブルAのグループとBタイプの照合用のテーブルBのグループがSDRAM11の上にある。Aタイプの照合処理の第1段テーブルA-1は0番目のキャッシュエントリにキャッシングされるアドレスに確保される。Bタイプの照合処理

の第1段テーブルB-1も同様に0番目のエントリになるアドレスである。これらはキャッシュエントリが複数セットで多重化されているため、同時にオンキャッシュとなりうる。

- 同様にAタイプの照合処理の第2段テーブルA-1は1番目のキャッシュエン
- 5 トリ1、Bタイプの照合処理のB-2には2番目のキャッシュエントリが対応する。この場合、複数の照合がキャッシュエントリで多重化されず、それぞれ独立したキャッシュエントリが割り当てられている。キャッシュが4セットであれば、A-2であれば全16テーブルで1番目のキャッシュエントリの4セットを所有し、最新の4テーブルがオンキャッシュである。下位段、特に、最終段のテーブル
- 10 A-8およびB-8は数が多くなるので、1段に対して複数エントリを割り当てることも有効である。

- 図55に、空テーブル管理キュー69が、メモリ11の上でテーブル28を管理する一例を示してある。検索種別毎にあるいはテーブル段毎にキャッシュエン
- 15 トリを分散配置させることでオンキャッシュ率を高める。そのためには、テーブル28を管理する側、すなわち管理キュー69が分類装置50のテーブル確保要求に応じて適切なテーブルを割り当てる必要がある。空きテーブルキュー69は空きテーブルを検索種別（レイヤー1）とそれに含まれるテーブル段位置（レイ
- 20 ヤー2）の2階層に分類した個別キューで管理する。分類装置15の初期化時に、キャッシュエントリ毎にテーブル28をグループ化し、更にROM等に設定された（あるいは外部からコンフィグレーションとして与えられる）、（検索種別数×テーブル段数分の）空きテーブル個別キューとキャッシュエントリの対応に従い、テーブルをキューにつないでいく。

- 図56に空きテーブルキュー管理機構69の応答を示してある。空きテーブル
- 25 キュー管理機構69は、検索テーブルツリーへのエントリ登録時に照合装置50から与えられるテーブル確保要求（ステップ211）に応じて、空きテーブルを取り出し応答する。この際、検索種別とテーブル段位置が必要となるため、要求情報としてそれらを要求パケットに持たせることとする。管理機構69は、要求情報の検索種別（レイヤー1）・テーブル段位置（レイヤー2）に対応する
- キューから空きテーブルを1個取り出し（ステップ212）、応答としてそのア

ドレス（タグ情報）を照合装置 50 に返す（ステップ 213）。

図 15 に示したメモリ共有型の分類装置 15 において、SDRAM 11 と照合装置 50 の間にデータキャッシュ 17 がある場合、データキャッシュ 17 の一つがミスヒットするとデータキャッシュ 17 および SDRAM 11 はこのアクセスに占有されて他のアクセスは待たされる。図 16 に示したメモリ分離型の分類装置 15 において、SDRAM 11 と照合装置 50 の間にデータキャッシュ 17 がある場合も、同様に、照合装置 50、データキャッシュ 17 および SDRAM 11 はこのアクセスに占有される。これらはデータキャッシュ 17 と SDRAM 11 が同期的（シリーズ）に接続しているためである。そのため、1 つ遅れると、その遅れは後段に影響する。パイプラインの効率を上げるには、全てのステージにかかる時間を同一にしてやらねばならない。しかしながら、SDRAM アクセスが入ると難しい。

これに対し、要求応答方式のデータキャッシュ 17 を採用することができる。すなわち、検索対象キー 21 に対して入力順に照合結果が出力されるというパケット順序の維持はバックトラックが発生することを前提とした段階で不可能である。このため、識別情報（ID）32 を導入することにより順序の問題はクリアしている。したがって、ある検索対象キー 21 の照合のためにテーブル 28 が要求されたときに、オンキャッシュでなければ照合装置 50 にその旨を通知する。照合装置 50 は、処理を中断し、待ち行列に戻し、次の検索対象キー 21 のパケット 33 を入れる。データキャッシュ 17 は SDRAM 11 へアクセスする。一方、照合装置 50 からテーブル 28 を要求されたときに、オンキャッシュならば照合装置 50 にテーブル 28 を渡し、照合装置 50 が処理を進める。さらに、照合装置 50 において、照合タグ情報 34 を先読みしてキャッシュシステム 17 に渡し、必要なテーブル 28 をプリロードするシステムは有効である。さらに、照合装置 50 が照合のためにテーブル 28 をバッファ 59 にロードした時点で、次の照合タグ 34 を入れてオンキャッシュを確認することもある。

このように、本例のキャッシュシステム 17 においては、検索種別および照合段階という要素でキャッシュラインの排他性を高めてオンキャッシュ率を高めている。従来の CPU でのデータキャッシュにおいては、キャッシュラインサイズ

をテーブルサイズに一致させるという発想はリニアアドレス空間を否定するので従来のキャッシュシステムからは得られないものである。さらに、照合テーブル各段を異なるキャッシュエントリに割り当てるという発想も、論理アドレス（のインデクス部分）とテーブルの割付決定を実行時に行うことであり、OSの元ではCPUで実行することは極めて難しい。CPUでは論理アドレスとテーブルの範囲割付決定をコンパイル時に行い、実行時には決められた範囲内から1テーブル分を確保するだけである。この実行時メモリ割付支援機構が認識するのは先頭論理アドレスとサイズである。動的メモリ割付は確保、開放を繰り返すとメモリに隙間を生じ、これをガベージコレクタで整理するというのが前提であった。従い、インデクス部分の衝突を避けて、キャッシュ効率を上げる仕組みを入れ込めなかった。

これに対して、ここで説明している照合方法および照合装置においては、テーブルビット長を固定化（上記の例では4ビット）することによってテーブルを標準化した。このため、テーブルサイズとキャッシュラインサイズとを一致させることが可能になった。そして、キャッシュメモリ17やSDRAM11に対して検索種別および照合段階で排他的なエリアを割り付けたとしても、テーブル同士は照合タグ情報によりリストでつながれている。このため、エントリの（実行時の）削除、登録に対応することができ、オンキャッシュ率を高めるために排他的な領域設定をすることが他の処理において障害となっていない。

20

履歴キャッシュ付き分類装置

図57に、履歴キャッシュ18を備えた分類装置15の概略構成を示してある。照合装置50の前に履歴キャッシュ18をおくことで、検索処理をさらに高速化でき、検索処理の遅延時間を短縮できる。説明してきたように、本発明の照合方法および照合装置、さらに、分類装置および検索装置は、簡易なハードウェアでビット長の増加にも対応でき、照合速度も早く、さらに、汎用メモリを利用して十分な処理速度が得られるといった多くのメリットがある。しかしながら、複数の照合装置をパイプライン状に接続するか、あるいは照合装置を繰り返して使用することにより照合結果を導く。このため、同一の検索対象ビットストリングが

高頻度で表れるようなジョブにおいては、CAMあるいはTCAMと比べたときにレイテンシがあるという差がある。したがって、レイテンシが重要となる検索においては、履歴キャッシュ18を用いるメリットがある。

図57に示した履歴キャッシュシステム18は、SDRAM共用型履歴キャッシュシステムであり、照合テーブル28の置かれるSDRAM11を共用する構成である。図58に示すように、キャッシングアルゴリズムとして「8ビット化ハッシュ+LRU」を使用することができる。そのため、本例のキャッシュシステム18は、ハッシュテーブル検索器18aと、ハッシュ用のキー一括比較器18bとを備えている。ハッシュテーブル18cは照合装置50と同様に4ビット(16エントリ)×2段のツリー構造で構成し、照合装置50と履歴キャッシュ18の間でデータキャッシュ利用方法の整合性を取るようになっている。

履歴SRAMキャッシュ付き分類装置

図59に、履歴キャッシュ専用のSRAM13を備えた分類装置15の概略構成を示してある。履歴キャッシュ18のメモリを専用化した構成であり、テーブル28を記憶したSDRAM11と分離することにより、照合装置50との競合を防止して照合速度の向上を図る。それと共に、ZBL-RAMやZBT-RAM等の応答性のよい低遅延SRAMを、履歴キャッシュ用に占有し、履歴キャッシュ18の性能を向上する。キャッシングアルゴリズムは「8ビット化ハッシュ+LRU」を使用できるが、共用型の場合は照合テーブルとの整合性のあるメモリブロック構成が望ましかったのに対し、図60に示すように、専用SRAM型の場合は、メモリビット幅やエントリの構成を履歴キャッシュに合わせて最適化することができる。

論理演算装置(AND化器)の構成

上述した検索装置10あるいは分類装置15は論理演算装置(AND化器)19を備えている。この分類装置15は、汎用型なので、ルートタグ31により検索種別を上流の制御装置あるいはアプリケーションによって設定することができる。したがって、1つまたは複数の分類装置15による検索種別の異なる検索結

果（１次元範囲の検索結果）を複数用いて、AND化器１９で論理積を演算することにより、多次元範囲の検索を行うことができる。本例のAND化器１９においては、１次元範囲の検索結果を集合として取り扱い、行列演算により多次元範囲の検索結果を単純な操作で得られるようにしている。

- 5 その一例がSPD検索機能８である。１つのパケットデータ ϕp を管理するために、ソースIP、デスティネーションIP、ソースポート、デスティネーションポートおよびプロトコルを示すビットストリングを少なくとも照合して、それらの全てを満足するルールを検索する際に、それらを示すビットストリングを検索対象のビットストリングとして、それぞれ異なる分類装置１５において照合する。
- 10 それぞれの分類装置１５により得られる複数の分類結果は、範囲検索により複数のルールが含まれたものとなり、それらの分類結果をAND化器１９において論理積を演算することにより、パケットデータ ϕp を管理するルールを抽出することができる。

- 15 まず、集合Aが要素 $\{A_1, A_2, \dots, A_n\}$ で構成されたとする。要素 A_i がバイナリ（０／１）の m bit列“ $a_{i1}, a_{i2}, \dots, a_{im}$ ”で表せるとする。同様に集合Bが要素 $\{B_1, B_2, \dots, B_n\}$ で構成されたとする。要素 B_i がバイナリ（０／１）の m bit列“ $b_{1i}, b_{2i}, \dots, b_{mi}$ ”で表せるとする。

- 20 このとき、集合A、Bの積（AND）集合 $A \& B$ は、行列におけるA、Bの積「 $A \times B$ 」で表せる。詳しい演算式は、以下の通りである。

$$A_i \times B_j = \sum_k a_{ik} * b_{kj} \quad \dots (4)$$

- 25 ここで、 Σ 内はブール乗算（*）とブール加算（+）に展開される。ブール乗算・ブール加算の定義は、ブール乗算「*」はENOR（Exclusive NOR、非排他論理和）、ブール加算「+」はANDである。ブール加算は、（ $1+1=1$ $1+0=0$ $0+1=0$ $0+0=0$ ）である。また、ブール乗算は、（ $1*1=1$ $1*0=0$ $0*1=0$ $0*0=1$ ）である。ENORは比較回路の「=」のロジックと同じであるので、 $A_i * B_j$ は比較等号回路を「AND」で

結んだものだと理解してもかまわない。行列で理解したほうが見通しは良い。

図6 1に、集合Aが2要素、集合Bが3要素、さらに各要素が3ビットで現れる例を具体的に示してある。左辺行列Aは集合Aの要素A 1, A 2が各行に並べられている。また、行列Bは集合Bの要素B 1, B 2, B 3が各列に並べられている。右辺行列Zの1行1列は「 $a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31}$ 」であり、「 $A_1 \times B_1$ 」である。これはAND化器19におけるブール演算の定義により、A 1とB 1のビット比較結果（1＝一致／0＝不一致）を意味する。同様に行列Zを列方向で見ると、1列目は「A 1とB 1の比較結果」と「A 2とB 1の比較結果」になっている。同様に行列Z 2列目は「A 1とB 2」と「A 2とB 2」の比較結果である。つまり、行列Zの各列は行列A全体（集合A）と行列Bを構成する各列（集合B要素B i）の比較結果である。

したがって、行列演算の結果、例えば、右辺行列Zが図6 2に示すようになっていけば、左辺行列Xの{ a_{11} a_{12} a_{13} }行（集合要素A 1）が集合Aと集合Bの積集合A&Bである。つまり、行列Zの各行において行内に1つでも1が立つ行に対応する行列Xの行が積集合である。ここで「1つでも1が立つ行」を言い換えれば、行列Zの各行において行内の値の論理和ORを取った結果が1となる行に対応する行列Aの行が積集合である。

したがって、行列Aに対して、論理積を求める要素を掛け算し、1つでも1が立つ行が残れば、それらの要素の積集合が得られたことになる。図6 3にLSI化を念頭に行列演算を一般化したものを示してある。左辺の行列Xは行毎に集合Aの要素A 1, A 2, ...である。左辺の行列Yは列毎に集合Bの要素B 1, B 2, ...に加え、集合Cの要素C i, ...で構成される。列の数は限定されない。行列Zを列毎にみると、行列X全体と行列Yのある1列との演算となっている。行列Zは最終的には行毎に行内で論理和ORが演算され、 $A * B * C * \dots$ の積集合要素が決定される。従って、行列Zはその値全体を保持しておく必要はなく、列毎にブール行列演算した上で、累積結果としてOR演算していけばよい。

図6 4に、AND化器19のハードウェア構成の一例を示してある。AND化器19は、複数のブール行列演算レジスタ19aを有しており、集合Aの各要素

- が各レジスタ 19 a にセットされる。さらに、各々のレジスタ 19 a に対応して、レジスタ 19 a の演算結果の累積論理和 (OR) を演算して結果を蓄積するアキュムレータ 19 b を備えている。各アキュムレータ 19 b は「0」に初期化される。このAND化器 19 に対し、入力要素 19 i n として、集合 B の要素 B i 、
- 5 集合 C の要素 C i などの各列を順次に投入すると、入力列毎に以下の演算処理がおこなわれる。まず、ブール行列演算レジスタ 19 a と入力列による、ブール行列演算 $A_n \times Y_i$ (ENOR 演算、AND 演算となる) が行われる。次に、ブール行列演算の結果とアキュムレータ値による OR 演算が行われる。そして、OR 演算の結果によりアキュムレータ 19 b の値が更新され、最終入力後にアキュム
- 10 レータ 19 b に 1 が残る行に対応する集合 A の要素群が集合 A, B, C, ... の積 (AND) 集合、すなわち、 $A \& B \& C \& \dots$ である。

- ブール行列演算は ENOR 論理によるビット比較を AND 論理で繋いだものであり、回路的に見れば、各ブール行列演算レジスタはレジスタ値 A_n と入力値 Y_i との m ビット完全一致比較結果を出力し、その比較結果をアキュムレータで累
- 15 積 OR していることになる。ブール行列演算レジスタの数やビット長にもよるが、ENOR 論理および AND 論理を組み合わせる回路を作る事は極めて簡単である。そして、入力値 1 個とレジスタ値複数個が一度に比較できるので、レジスタ数分の並列性がある。したがって、本例の AND 化器 19 は、単純な構成でありながら、短時間で多次元範囲検索の結果を導出することができる。あるいは、並列レ
- 20 ジスタがコスト高という場合は、レジスタを 1 個あるいは数を減らして、レジスタ値はメモリにし、入力値に対して 1 行分ずつ繰り返しブール行列演算することによっても本例の AND 化器 19 は実現できる。

- レジスタ値をメモリに展開することにより、メモリの許す限り、ルールを入れることができるというメリットがある。このため、多数のルールあるいは要素を
- 25 有する集合から論理積により要素を絞り込むのに適している。ただし、繰り返し処理でありメモリリードもあるため、処理時間は長くなる。ブール行列演算処理とメモリリードをパイプライン化することは可能であり、この構成によりメモリリード処理時間を隠蔽することも可能である。

さらに、AND 化器 19 は内部にアキュムレータ 19 b を持ち、またレジスタ

1 9 a に最初の集合の要素を保持する構造である。このため、複数の多次元範囲
検索の並列実行を考えるとそれらは共有資源である。したがって、何らかの資源
排他占有が求められる。一方、分類装置 1 5 においては、複数検索の各次元の検
索が並列におこなわれえるため、例えば、検索系 S_1 , S_2 の各次元 A, B につ
5 いて、分類器からの出力は S_1A , S_2B , S_2A , S_1B のように検索系も同検
索内の次元も順序性のない出力となりうる。先に分類装置 1 5 から出力された検
索系が AND 化器 1 9 を占有利用する方式でも、先に分類装置 1 5 からの全次元
出力がそろった検索系が占有利用する方式でも、いずれにせよ分類装置 1 5 の出
力が非同期・無順序であるため、AND 化器 1 9 の利用率および多次元範囲検索
10 としてのスループットを高効率でおこなうためには、AND 化器 1 9 も並列化す
ることが望ましい。AND 化器 1 9 のレジスタおよびアキュムレータをメモリ化
した場合は、メモリ上での並列化でも効果が得られる。

また、AND 化器 1 9 の動作を考えたとき、初期設定「BEGIN (レジスタ設
定・アキュムレータ初期化)」と終了処理「END (アキュムレータからの読み出
15 し)」が必要であり、同系内でも分類器入力と出力の間での順序性がない。この
ため、次元 A だから初期設定処理、次元 B だから終了処理というように操作は固
定的に決定できない。分類装置 1 5 からの出力を検索系ごとにキューイングし、
全次元の分類器出力が完了したのち、明示的に初期設定と終了処理をコントロー
ルする方式となる。結局、これらの制御もあるので、分類装置からの出力がそ
20 ろった検索系が AND 化器 1 9 を排他占有する制御方式が望ましいといえる。

AND 化器 1 9 の並列利用のためには、速度を要求されるなら AND 化器 1 9
そのものを複数用意することで対応できる。次善策としては、レジスタ 1 9 a お
よびアキュムレータ 1 9 b のメモリ保存によるメモリ上での並列化をおこなうこ
とで対応できる。さらに、履歴キャッシュ 1 8 を導入している場合、多次元範囲
25 検索における履歴キャッシュ 1 8 の使い方を工夫することで、AND 化器 1 9 の
使用率を下げることができ、AND 化器 1 9 を並列利用しなくても十分な処理速
度を維持することも可能である。例えば、32 ビット 5 次元の多次元範囲検索で
あれば、160 ビット (32 ビット \times 5) のキーによる履歴キャッシュ 1 8 を採
用し、履歴キャッシュ 1 8 のエントリデータとして、初回検索時に AND 化器 1

9を使って演算した5次元の積集合結果を登録する。つまり、履歴キャッシュ18には分類装置15の最長一致検索の履歴、完全一致検索の履歴、1次元範囲一致検索の履歴をエントリデータとして登録することも可能である。しかしながら、多次元範囲検索の場合は、各次元個別の履歴ではなく、全体としての多次元範囲検索の履歴をエントリデータとして取り扱うことによりAND化器19による処理がクリティカルパスになることを抑制できる。

上記に、幾つかの例を示しながら本発明について詳しく説明しているが、上記の例に本発明が限定されるものではない。

10 省略テーブル

以下では、照合テーブル28の数をさらに削減する方法を説明する。照合装置50を用いた検索においてはエントリビット列を4ビットずつの部分エントリに分割してテーブル化し、テーブル化ビットツリーにより登録し、検索していく。この方法では、ある段以降が1エントリしか存在しなくてもテーブルを生成し、検索においてもテーブルを辿る必要がある。これを改善し、ある段以降が1エントリであれば下位テーブルを省略するテーブル構造を導入することが可能である。

図65に下位テーブルを省略する例を示してある。テーブルBは本来のテーブル構成では「B-C-D」となる1エントリが存在するが、下位が1エントリであるため「C-D」を省略し、その代わりにテーブルBに省略情報（エントリの省略した下位ビット列、省略段数、省略マークなど）を記録しておく。テーブルEでは直接の下位接続はテーブルFの1個であるが、エントリとしては孫テーブルで分岐したGおよびHの2エントリ以上が存在するため、省略はされない。

テーブルBの下位に収容される2エントリ目の登録時に、省略された下位テーブルは展開（生成）される。上図では「B-C-X」と繋がるべきエントリXの登録が行われる際、テーブルBにて下位テーブルの省略情報を検出し、第一エントリに相当するテーブルCおよびテーブルDの生成し、第二エントリに相当するテーブルXの生成をおこなう。

ここで、第一エントリのテーブル生成と第二エントリのテーブル生成はシリーズにおこなうのではなく、テーブルBの次の段は同ビット列なので共通テーブル

Cを生成するという方法もある。テーブルCの次では異なるビット列なので分岐しテーブルDおよびテーブルXを生成するというように、2エントリが分岐する段までは統合的にエントリを生成することができる。したがって、エントリを登録する処理時間を短縮できる。

- 5 また、分岐の次段では再度それぞれの下位が1エントリになる。このため、そこで省略テーブル形式でテーブル生成することになる。例えば、上図においてテーブルDおよびテーブルXが最下段ではなく、さらに下位に延びているのであれば、テーブルCで分岐した次のテーブルDおよびテーブルXはその下位にそれぞれ1エントリしか存在しない。このため、下位テーブルは省略される。省略
- 10 テーブル形式で生成されると、検索時は照合装置50がテーブル28の省略マークを見て、そのテーブル下位が省略されていることを検出する。したがって、照合装置50は、次段のテーブルを読み出さずに最終照合結果を得ることができるので、照合時間も短縮できる。

- 15 圧縮テーブル（照合テーブル）28が省略テーブルの場合には、4ビット×n段分のエントリ省略ビット列が記録される。したがって、照合装置50では、エントリビットをシフト読み出ししながら検索モードに応じた照合を行うことが可能となる。1エントリに対する完全一致・近似一致の照合であれば、上述した照合器50の比較部52を使う代わりに、専用化した比較器を実装したほうが効率的な場合がある。また、専用比較器を使用するのであれば、4ビット比較×n回
- 20 でおこなう必要はなく、リソースのバランスを見た上で多ビットの比較器とすることが望ましい。

- 25 省略テーブルでは検索対象キー21と1エントリとの比較であるため、完全一致検索であれば完全一致照合、範囲検索であれば、大・同・小が判定できる比較照合が可能な比較器でよい。また、比較後のエントリ選択処理は不要であり、比較結果から検索結果を判定するだけである。完全一致であれば比較で一致なら検索成功、不一致なら検索失敗である。範囲一致であれば比較で大または一致であれば（右倒れ型）検索は成功、左倒れ型の場合も同様に比較が小または一致なら検索成功である。

削除については、削除の結果、1エントリとなる部分を検出し、その下位から

- エントリビット列を集約し再度の省略テーブル化を行わなければならない。テーブル管理の処理時間は増加する可能性がある。しかしながら、上述したように、照合時間を短縮できるので、メリットは大きい。各段のテーブル28に収容エン
- 5 トリ数カウンタを設け、上位から下位へ削除エントリを辿る際にカウンタを減算し、1エントリのみとなるテーブル位置を検出する。そのテーブルを省略テーブルに変更し、その下位の省略対象エントリのテーブル列を辿り、最下段から上位へと省略ビット列(=エントリNo)を合成しながら上位へと上げていき、省略
- 10 テーブルに集約して省略ビット列として記録するという方法がある。削除によるテーブル再省略化は、上述した更新処理よりも複雑でありテーブルアクセスも増える(時間がかかる)。従って、登録時のみ省略・展開をおこない、削除では再省略化をおこなわない方式もありうる。検索頻度と登録・削除頻度とのバランスで、登録削除が一定以上多い場合、展開・省略の繰り返し、すなわち、テーブル確保・開放の繰り返しとなり、分類器全体の性能低下を招く可能性もある。

15 照合装置のパイプライン化

- 図66に、要求応答方式のメモリアクセスと照合装置の内部のパイプライン化の概要について示している。上記においても説明したが、照合装置50においてはマスクデータ28aの読み出しがボトルネックの1つである。マスクデータ28aがなければ、判定ユニット54から下流の処理(VLDからSELおよび出
- 20 カユニット55における処理)が実行できず、マスクデータ28aの読み出しと並列に実行できるのは主に比較ユニット52の処理だけである。従って、照合器50の内部処理もパイプライン構成とし、テーブルロード(マスク読み出し)フェーズにてキャッシュ(SDRAM)へメモリリード要求を出し、応答のあったものからVLD以降の処理をおこなうことが考えられる。そのために、VLD
- 25 ステージの前に待ち行列57があり、そこで応答の待ち合わせがおこなわれる。

図66に示した例では、メモリ要求～応答の間にCMPステージ52を入れているため、待ち行列57に待ち合わせ機構が必要となる。データキャッシュなしの場合は、メモリアクセスはFIFOであり待ち行列57もFIFOによる待ち合わせでよいが、データキャッシュがあった場合、ヒット・ミスヒットにより順

序性がなくなり、CAM等のFIFO型よりは高度な待ち合わせ機構が必要となる。データキャッシュなしの場合でも、メモリアクセスをとまなわないモード（スルーなど）があるので、やはり単純なFIFO待ち合わせ型の行列では無理がある可能性がある。メモリアクセスを伴わない場合も、ダミーのアクセス要求を出してダミー応答を戻させ、応答数とFIFO待ち数を整合させればFIFO型で対応できる可能性がある。

図67に異なる照合装置50の例を示してある。この例では、メモリアクセスを伴うものとそうではないものを分岐させ、メモリアクセスを伴うものは非順序の応答をVLD直前の合流待ち行列57に戻す。一方、メモリアクセスを伴わないものは、待ち行列57の前のステージ57pで必要な処理があればおこない、合流待ち行列57に入る。この場合、合流待ち行列57は単なる2方向からの合流のみであり待ち合わせは不要となる。この構成では、待ち合わせ機構を不要とするため、メモリアクセスとCMP処理52の（空間的な）並列化はなされないが、照合装置50の内部のパイプラインにより時間的な並列化には有効である。

マスクデータのプリフェッチ

図68に、マスクデータを先読み対応可能なテーブル化によるメモリアクセストールを解消する例について示してある。マスクデータの読み出しに係るボトルネックを解消するためにテーブル28の構成を変更する。図68(a)に示すテーブル28は、そのテーブルのエントリ有無を示す情報としてマスクデータ28aがあり、次テーブルのアドレスを示すアドレステーブルを示すNextTAG×16エントリで構成されている。図68(b)に示す第1型の改良型テーブル281は次段のアドレス「NextTAG×16エントリ」28bの次に、次段のマスクデータ「NextMask×16エントリ」28pを配置したテーブル構成となっている。また、図68(c)に示す第2型の改良型テーブル282は、次段のマスクデータ「NextTAG」28bと次段のマスクデータ「NextMask」28pのペアが16エントリ登録されているテーブル構成である。いずれも、次段の照合テーブル28のアドレスだけではなく、マスクデータ28aも備えているので、次段のマスクデータ28aをプリフェッチすることができる。

ただし、改良型のテーブル 281 および 282 においても、テーブル管理や検索以外での利用のため、次段のマスクデータ「NextMask」と同じ値が、次段のアドレス「NextTAG」の指すテーブル 28b に記録されていることが望ましい。したがって、あるテーブルにはそのテーブル自身のマスクデータと、次段のマスクデータの 2 種のマスクデータが存在することが望ましい。テーブル下位からエントリを辿っていく場合に、上位のテーブルにしかマスクデータがない場合、そのテーブルだけでは隣接エントリへの移動が不可能になってしまうからである。

次段のマスクデータをプリフェッチする構成にするのに伴い、照合装置 50 の間をフローする照合継続情報のデータパケット 33 にマスクデータ 28a を追加する。この改良により、照合装置 50 の処理においては、入力パラメータ（照合継続情報）33 としてその照合装置 50 で使用する照合テーブルのマスクデータ 28a が与えられる。このため、マスクデータ 28a を必要とする判定ユニット 54 から下流の処理と、ロードユニット 53 の処理とを並列に実行することが可能となる。

マスクデータ 28a をリードすることに起因するストールを解消するという点では、分類装置 15 あるいは照合装置 50 に内部 SRAM を設け、各テーブルのマスクデータ 28 を照合装置 50 の内部に配置することで並列度を向上することも可能である。

以上に説明した照合方法および照合装置は、IP ヘッダのようなビットストリングを一致検索したり、範囲検索してルーティングするデータ管理方法を実現するのに好適である。すなわち、照合装置 50、および照合装置 50 を備えた分類装置 15 は、データ管理装置 1 の検索機構 10、8s、7s および 6s として好適なものである。しかしながら、本発明の照合方法および照合装置の適用範囲は、これらに限定されるものではない。

MIN 検索・MAX 検索

複数の照合装置 50 を備えた分類器 15 を様々に応用する場合、MIN 検索・MAX 検索があると便利である。バイパスを MAX バイパスとしていた場合に、検索対象キー {11...11} で右倒れ近似検索をおこなうことで、登録エン

トリの中からMAXエントリを検索することはできる。だが、MINエントリは検索できない。従って、この場合、MIN検索モードで検索することでMINエントリを取り出すことになる。ソートなどのように検索ではなく、登録&取り出しの処理において、分類器15を利用できる場面がある。このとき、MIN検索およびMAX検索の両方を備えていることはメリットがある。

照合器50において、比較ユニット52の処理を省略し、判定ユニット54からなる検索モードが実行できるようにすることも有効である。たとえば、比較ユニット52のCMP iからは常に「=」(01)が出力されるようにする。あるいはMIN検索・MAX検索モードになったときには、比較ユニット52をスルーして判定ユニット54のVLD 54 aに「01」が入力される回路としても良い。

この検索モードにおいては、VLD 54 aにより有効エントリのみ「01」となり、SEL部54 bにて、MAX検索モードであれば比較結果が「01」のエントリの中から最もエントリNoの大きなエントリが選択される。また、MIN検索モードであれば最もエントリNoの小さいエントリが選択される。右倒れ・左倒れの G_L ・ G_R 関連部でエントリ選択論理を「 $> (11)$ ・ $< (10)$ 」から「 $= (01)$ 」に変更することにより実現できる。テーブル乗り移り処理であるタグ更新部55 bは完全一致検索モードに準じ、出力 S_r ・ S_l で指定されたエントリNoを次テーブルとした照合タグ情報34の更新をおこなう。以上により、各段のテーブルで最小エントリ・最大エントリを辿っていくMIN検索・MAX検索機能を提供できる。

次に、分類装置15をソートへ適用する場合、ソート後のデータ読み出しのための拡張が必要となる。昇順先頭読み出し・降順先頭読み出しはMIN検索・MAX検索で可能である。従って、昇順次読み出し・降順次読み出しのためのコマンドが必要となる。これを実現するためには、テーブルは双方向リンク型テーブルとする。また、昇順先頭読み出し(MIN検索あるいは{0...0}近似検索)・降順先頭読み出し(MAX検索あるいは{1...1})の出力パラメータとして、最下段テーブルのTAG+エントリNoを読み出しハンドラとして出力する。

さらに、コマンド「NEXT・PREV」を設け、読み出しハンドラ（最下段テーブルTAG+エントリNo）を入力パラメータとして受け取ると、次エントリの結果TAGと共に更新した読み出しハンドラ（最下段テーブルTAG+エントリNo）を出力する。照合器50の内部では、まず照合TAG（読み出しハンドラの最下段テーブルTAG）34に従い、テーブル28よりマスクデータ28aを読み出す。読み出しハンドラのエントリNoに基づき、次エントリNo（前エントリNo）を算出し、テーブル28から対応するエントリのアドレス「Next TAG」（最下段なので結果TAG）を読み出す。もし、次エントリNo（前エントリNo）が別のテーブルになる場合は、双方向リンクの上位リンクを使い1段上位テーブルに遡り、そこで隣接エントリに移る。1段上位にも自エントリしかなかった場合は更に上位に遡っていく。上位テーブルで隣接エントリに移ったら再び最下段まで降りなければいけないが、これはMIN検索・MAX検索モードによりおこなう。

15 データベース検索

この分類装置15は、リレーショナルデータベース（RDB）にも適用できる。バイナリビットツリーによる検索はデータベースではインデクスに相当する。したがって、複数のフィールドをバイナリビットツリー登録しておけば、これらをキーとしてデータベースを検索できる。検索の種類は最長一致でも、範囲一致でも、完全一致でも対応できる。この処理では、複数のフィールドを検索後のAND処理（積集合をとる処理）を高速で行うことが望ましい。つまり「A0<身長<A1」と「B0<体重<B1」を別々に検索できたときに、「A0<身長<A1」かつ「B0<体重<B1」の集合を高速に得られることが望ましい。予めフィールドが固定的に決定されているならば、そのフィールド群による複合（多次元）インデクスを生成しておくのが一般的である。対象フィールドが固定的に決定できず、複合インデクス（身長と体重の2次元インデクス）が生成できない場合、「CPU+OS」の検索システムではおそらく次の方法で論理積を演算する。すなわち、レコード名をRiとすれば、「A0<身長<A1」の集合{R1, ..., Rm}（個数はm個）と、「B0<体重<B1」の集合{R

1, ..., $R^{\prime}m$ } (個数は m^{\prime} 個) を求め、二つの集合のメンバを総当りで比較し両方に属するメンバを残す。これがAND集合になる。この場合、比較計算は $m \times m^{\prime}$ 回になる。

- 5 1, ..., Rm } を分類器15に登録し、集合 $\{R^{\prime}1, \dots, R^{\prime}m\}$ を照合した時、前の集合で登録したメンバにヒットすればこれがAND集合のメンバになる。この場合、比較計算は $m + m^{\prime}$ 回になり、処理速度が向上する。

- この分類装置は、積集合 (AND集合) だけでなく、和集合 (OR集合) の演算も、ユニーク化も、また、ソートも簡単にできる。「CPU+OS」の検索システムで、データベースにテンポラリーな印をつけて良いならば、次の方法で論理積 (AND) を演算できる。集合 $\{R1, \dots, Rm\}$ の各メンバを検索の都度、印をつける。 $\{R^{\prime}1, \dots, R^{\prime}m\}$ の各メンバを検索の都度、前の印があるものをあげる。この場合、比較計算は $m + m^{\prime}$ 回になるが、不要な印の処理やディスク本体のデータベースへのアクセスなどの問題が起こり、時間がかかりすぎる。ビットマップ法は高速に論理積 (AND化) できるがルール数に制約があるなど汎用化は難しい。
- 10
- 15

- これに対し、上述した分類装置によるAND集合をとる方法は、パケットフィルタリング等の複数ルール集合のAND集合化にも使える。本分類器15では、例えば、5つのID (5種類のビットストリームに対する検索) に対応する5つのインスタンスを並列に動かし、ID毎にヒットするルールに印をつけていく。既に4つの印を確認したインスタンスはこれをANDインスタンスへ通知する。ANDインスタンスは通知される都度、今の最優先ルールと比較し、更新する。5つのインスタンスが全て終了した後、今の最優先ルールに決定する。
- 20

25 非リニアメモリアドレッシング

本発明の分類装置により、ツリー構造型の非リニアメモリアドレッシングの実現も可能となる。アドレスを検索対象のキーとし、本発明の分類器をメモリ管理機構に取り込むことで、上位層からみて柔軟なアドレッシングが行えるメモリ空間が実現できる。これは従来のリニアアドレス空間メモリに対し、上位層が明示

的に構造化管理を行っていた部分をメモリ管理機構が隠蔽カプセル化し、メモリを抽象化オブジェクト化することに他ならない。

さらに、従来、CPUのコードでおこなっていたメモリのオブジェクト化アプローチとは異なり、ハードウェアベースのアプローチとなる。したがって、高速および効率的に、オブジェクト化を実現できる。特に、検索のためのメモリ利用
5 の場合、分類器によるアドレッシング（＝照合）が効果的に利用できる。

図69に、その一例を示してある。受信時、インターフェイス（I/F）230を経由してリニアアドレス（論理アドレス）の書き込みデータ（受信データ）239を受け付ける。分類器15は、受信データ239をSDRAMメモリ11
10 へ書き込む。分類器15は、アドレス変換機能232を持っているので、論理アドレス（リニアアドレス）から実アドレス（リストやツリーなど何らかの構造をもった非リニアアドレス）へ変換し、SDRAM11に書き込む。一方、I/F230を介してデータを送信している方は、リニアアドレスで書き込むことができる。送信時も同様である。I/F230からリニアアドレスで読み出すときに、
15 分類器15を経由することで、アドレス変換をおこなう。これにより、リニアアドレスから実アドレスへ変換し、非リニアなアドレスからの連続読み出しに対応できる。

CPU235も分類器15によるアドレッシングを利用できる。CPU235はデータフロー動作ではないため、分類器15が提供するアドレッシング配下の
20 メモリ11だけではコード実行等に支障がある。従って、コード蓄積やデータ加工用として独自にメモリ236を持ち、分類器15の配下のメモリ11はパケットメモリとして利用する。ルータ等を考えた場合、CPU235は分類器15によるTCP/IP処理を補助するものとして、ルーティングエントリ管理やフィルタリングルール管理などのハードウェア処理ではカバーできない処理を肩代わりする。つまり、分類器15がメイン機能を担当し、CPUはサブとして動作する。
25

テーブルルックアップ駆動プロセッサ

分類装置15は、状態遷移によるテーブルルックアップ駆動のプロセッサにも

適用可能である。図70は単純状態遷移を示しており、KEY (i) を検索し、その結果TAG (ルール) でKEY (i + 1) を読み取れるようにしておけば、「KEY (1) - KEY (2) - . . . - KEY (j) - KEY (1) 」のような単純ループ型の状態遷移ができる。

- 5 図71は、イベント連動型単純状態遷移を示しており、イベント (i) がトリガとなってKEY (i + 1) に遷移し、イベント (i) 以外でKEY (i) に戻る。KEY (i) はイベント (i) 待ち行列でイベント (i) を待つ。イベント (i) が来たらKEY (i) にイベント (i) を直列につないで検索する。イベント (i) は1ビットで、正が {1}、誤が {0} としKEY (i) を「10
- 10 1 . . . 01」とすれば「101 . . . 011」でヒットし、「101 . . . 010」でミスヒットとなる。ヒットルールにはイベント (i + 1) 待ち行列アドレス、KEY (i + 1) が書き込んである。ミスヒットルールにはイベント (i) 待ち行列アドレス、KEY (i) が書き込んであり、戻ることになる。さらに、本発明の分類装置は、条件分岐を含む状態遷移にも適用可能である。
- 15 さらに詳しく説明する。本書において説明している分類器における検索とは、「検索対象のキーを入力とし、そのキーをタグ (ルートタグ) で示されるテーブルツリーのエントリと照合し、結果 (結果タグ) を出力する」ことである。オートマトンのベースとなる状態遷移における検索は、「外部入力を現在状態からの遷移条件で評価し遷移先を決定する」と定義できる。ここで、「外部入力=検索
- 20 対象キー」、「現在状態=現在タグ」、「遷移条件=照合ロジックとタグの指すテーブルのエントリ記述」、とすれば、「外部入力 (キー) を現在状態 (現在タグ) からの遷移条件 (照合ロジックと現在タグの指すテーブルエントリ記述) で評価し遷移先を決定する」と定義でき、本発明によりオートマトンを実現できる。

- 上記においては、分類器あるいは照合器による検索は、ツリー状に組織化された構造 (テーブル) における上位から下位への一方向の遷移として説明している
- 25 が、状態遷移のように、ネットワーク状に組織化された構造における双方向の遷移であっても良い。これらの相違は、比較スケールが異なるだけとすることができる。すなわち、分類器の検索がツリーの一方向遷移であるというのはミクロの視点であり、状態遷移が双方向ネットワークであるというのはマクロの視点であ

る。1つの検索対象キーに対する動作という同一スケールで見た場合、照合器および分類装置は、予め定められたルール（エントリ記述）に従って、検索対象キーを評価し、結果を出力する。したがって、状態遷移においては、ある状態における遷移条件を、分類装置で検索する際の、検索のツリー状エントリ記述によって表現すれば良い。言い換えれば、状態遷移を検索の連鎖と見て、その状態毎の検索を分類器検索により実現するアプローチにより、分類器を、状態遷移マシン、あるいはオートマトンに応用できる。

図72（a）に、状態遷移を、分類器を用いた検索により実現する例を示してある。状態S1から条件「a」で状態S2に遷移し、条件「b」で状態S3に遷移する状態遷移マシンを仮定する。図72（b）に示すように、状態S1における状態遷移条件「a」および「b」を登録ビットパターン（エントリ）とする状態S1の登録パターンテーブル（S1テーブル）321を生成することができる。したがって、図72（c）に示すように、S1テーブル321から、照合用のバイナリビットツリー324を形成する複数の部分パターンテーブル325を生成する。そして、分類装置15により、現状態である状態S1で得られた状態遷移の評価元（条件）を検索対象のキーとして、上述した照合処理を行い、一致検索により遷移先の状態を出力できる。

図73（a）に、ネットワーク状に状態が遷移する簡単な例を示してある。この例では、上記に加え、条件「c」で状態S2から状態S1に移行し、条件「d」で状態S2から状態S3に移行する。また、条件「e」で状態S3から状態S1に移行し、条件「f」で状態S3から状態S2に移行する。図73（b）に示すように、状態S2における状態遷移条件「c」および「d」をエントリとする状態S2の登録パターンテーブル（S2テーブル）322と、状態S3における状態遷移条件「e」および「f」をエントリとする状態S3の登録パターンテーブル（S3テーブル）323とを生成することができる。状態S2に移行したときは、その状態における評価元（条件）が検索対象のキーとし、S2テーブル322から生成された部分テーブル325を用いて分類装置15により照合処理が行われる。状態S3に移行したときは、その状態における評価元（条件）が検索対象のキーとし、S3テーブル323から生成された部分テーブル325を

用いて分類装置 15 により照合処理が行われる。このように、移行した先で、照合する際に用いられる部分テーブル 325 を入れ替えることにより、ネットワーク状の状態遷移に対しても本発明を適応することができる。部分テーブル 325 の入れ替えは、部分テーブル 325 の最上位の部分テーブルのアドレスを示す
5 ルートタグを指定することで可能なことは説明した通りである。

図 74 に、分類装置 15 を用いたデータ処理装置 300 の一例を示してある。このデータ処理装置 300 は、状態により処理内容が決まるデータ処理回路 301 と、状態のそれぞれにおけるデータ処理回路 301 の処理内容（アクション）310 を格納した RAM 302 と、データ処理回路 301 の状態遷移エンジンとなる分類装置 15 とを備えている。RAM 302 には、状態に応じて分類装置 15 の検索に用いられる複数種類のテーブルツリー 324 と、それを構成する部分
10 テーブル 325 が記憶されている。このデータ処理装置 300 は、アクション実行部であるデータ処理回路 301 の状態が遷移する状態遷移マシンであり、分類装置 15 は、データ処理回路 301 から供給されたキー（検索キー）303 を検索対象のビットストリングとし、ルートタグ（検索テーブルツリー TAG）304 により RAM 302 に記憶されたテーブルツリー 324 を検索し、結果タグ 305 を出力する。この状態遷移マシン 300 において、検索テーブルツリー TAG 304 は、遷移条件テーブルツリー 324 のルートタグであり、結果タグ 305 は、次状態タグである。また、検索キー 303 は、状態遷移の評価元となる
15 データである。したがって、このデータ処理装置 300 においては、データ処理回路 301 は分類装置 15 に対して検索キー 303 を供給する検索対象供給手段としても機能している。

次状態タグ 305 とは、その状態への遷移に伴い実行すべきアクション記述 310 と、その状態から別の状態への遷移条件記述を指すタグである。実装によってはアクション記述や条件記述を、次状態タグ 305 に関連して決まる間接ポ
25 インタで示すことも可能である。アクション記述 310 では、遷移条件の評価元となる検索キー 303 を、データ処理回路 301 の内部あるいは外部状態から取り込むプロセスも指示されている。また、次状態タグ 305 により、次の遷移条件を検索する遷移条件テーブルツリー 324 が決まるように、アクション記述 31

0に、遷移条件テーブルツリー324を構成する照合テーブルの最上位を示すルートタグ304の情報を入れておくことも可能である。

したがって、このデータ処理装置300は、分類装置15において、照合装置50を用いた多段階の照合方法により状態遷移の評価元となるデータを検索対象のビットストリングとし、複数の状態遷移条件を示すデータを複数の登録ビットパターンとして照合し（照合する工程）、その照合結果によりデータ処理回路301の状態を遷移する（遷移する工程）制御方法により制御されている。このため、分類装置15に搭載される各々の照合装置50においては、上記において説明した照合方法における現段階を現状態として照合処理が行われる。そして、現状態の次の状態を示す照合継続情報が出力される。分類装置15において、次の状態を検索するために、1つの照合段階を費やしても良く、複数の照合段階を費やしても良い。

アクション記述310をコード、キー303を分岐先アドレス、分類装置15による遷移条件評価を分岐先メモリアドレス変換器と捉えることも可能である。したがって、データ処理装置300は、状態遷移ベースのプロセッサとして機能する。通常のプロセッサでは、CPUという状態遷移ハードウェア（チューリングマシン）の上にソフトウェアコードでアプリケーションとして状態遷移マシンを実現する。このデータ処理装置300は、分類装置15を用いて、直接的に状態遷移マシンを実現することができる。

さらに、この状態遷移マシン300により、有限状態オートマトン（FA）が実現できる。有限状態オートマトンは、入力により内部状態を変えていき、その結果として何らかの応答を出力する機械である。このデータ処理装置300においては、ルートタグ304を変更したり、アクション記述310の命令セットによりデータ処理回路301がRAM302の内容を書き換えたりすることにより、遷移条件テーブルツリー324（すなわち照合テーブル）を変えることができる。したがって、データ処理装置300は、内部状態の変更を、状態の遷移と状態ごとのアクションで定義した状態遷移マシンとして実現されている。

有限状態オートマトンには、決定性FA（DFA）と、非決定性FA（NFA）がある。決定性FAは、状態の遷移が一意に決まる有限状態オートマトンで

あり、非決定性F Aは、遷移状態が一意に決まらない有限状態オートマトンである。理論的にはN F AはD F Aに変換できるが、N F AをN F Aとしてハードウェアとして実現できることには状態空間の節約などの点で有意義である。「遷移状態が一意に決まらない」ことは、「状態が並立する」ということである。したがって、データ処理装置300により、並列データフロー型のシステムを構築すれば、その状態マシンではN F Aを実現することが可能となる。

通常のコピーでは、バックトラックなどにより、状態の並列を逐次方式で解決するため効率が良くない。しかしながら、並列データフロー型の分類器オートマトンでは効率的なD F Aが実現できる。

10

Q o Sへの適用

分類装置15をクラス階層構造データへのQ o Sに適用することができる。したがって、図1に示したルーティング装置1のQ o S機能7にも採用されている。IPルータ1におけるパケットφpのQ o Sに用いられるクラス検索を、IPアドレスを検索対象のビットストリングで行う場合、範囲検索を用い、検索テーブル7mへ登録するエントリを、クラス構造を反映したエントリとすることができる。

15

図75(a)に、クラス構造の一例を示してある。このクラス構造350に対し、図75(b)に示すような照合用のバイナリビットツリー354を形成することができる。このバイナリビットツリー354は、登録ビットストリングとして右倒れ型範囲検索用エントリが登録されている。したがって、図75(b)に破線で示すようなIPアドレス355を検索対象ビットストリングとし、分類装置15を用いた範囲検索により、そのIPアドレス355が該当する最下位クラスのエントリを得ることができる。この例では、IPアドレス355はクラス1.1に分類される。

20

25

分類装置15による検索で最下位クラスを分類する際に用いられるクラスツリー350も形式としてはテーブル型のツリー構造である。したがって、図76に示すように、クラスツリー350のテーブルと、検索ツリー354の部分テーブルとを関連付けすることが可能となる。このため、分類装置15においてテ

ブル照合を行う照合器50に、クラスツリー350に対する処理を行うハードウェアを追加することも可能である。例えば、自クラスの割り当てトークン消費、トークン不足の場合の上位クラスからの借り入れなどのために上位・下位ツリーを探索する機能を追加することが可能であり、クラスのトークン管理をハードウェア化することができる。

この管理は複雑になるが、上述した状態遷移（オートマトン）プロセッサを適用することができる。QoSのための検索やトークン管理のためのツリー探索などを、プロセッサとしてのアドレッシングとすることにより、データ構造そのものとデータへの処理コードが一体となったアーキテクチャが実現する。

図77に示すように、通常のCPU360では、メモリ361に格納された処理コード選択用検索テーブル362の検索処理がCPUコード363によりソフトウェア（アプリケーション）として実行される。このため、検索テーブルを検索するための検索コード（検索処理のソフトウェア）363と、処理コード

（データ処理のためのソフトウェア）364が分離している。したがって、CPU360は、検索コード363のロード（i段階）、検索コード363による実行（ii段階）、コンテキストスイッチ（データ処理コード364のロード）（iii段階）、処理コード364の実行（iv段階）となる。処理コード364のアドレス解決は、照合用の検索キー空間をソフトウェア的に検索することにより行われ、処理コード364のアドレッシング（コード空間）と検索キーのアドレッシング（検索キー空間）とは一致しない。

これに対し、図78に示すように、分類装置15を用いたプロセッサ300においては、検索テーブル（データ構造あるいはデータ空間）362に対して照合器50によるハードウェア検索で、検索キーに対応する処理コード364のアドレスを得ることで、検索キーから直に（ハードウェア的に）処理コード364のアドレスを解決することができる。そして、そのアドレス解決により得られた処理コード364を処理回路301により直接実行できる。したがって、コンテキストスイッチは不要となり、コード空間と検索キー空間とを一致させることができる。

通常のCPUで分類装置を用いたプロセッサのように直接的なアドレス解決が

行えないのは、CPUがリニアアドレッシングに基づくチューリングマシンであり、そのチューリングマシン上にソフトウェア的（仮想的）に状態遷移マシンやオートマトンを実装しなければならないためである。これに対し、分類装置を用いると、検索処理とアドレス解決とを結びつけ、状態遷移マシンあるいはオートマトンを直接的に実現することが可能となり、データ空間とコード空間を統合した効率的なシステムを実現することができる。また、この延長線上に、データとそのデータに対する処理手続きを統合したオブジェクト指向プロセッサのようなシステムも実現可能となる。

10 産業上の利用可能性

本発明に係る照合方法および装置は、完全一致検索、最長一致検索から多次元範囲一致検索までカバーでき、複数の照合装置を備えた分類装置は、パケットフィルタリング・SPD (Security Policy Database) 検索、ステートフルインスペクションなどのルール検索に用いられることができる。さらに、データキャッシュ、履歴キャッシュを導入することにより、照合速度を向上することが可能であり、多種多様な検索を伴う処理に適用できる。検索以外の応用機能としては、集合抽出、集合演算、整列（ソート）・ユニーク化、メモリアドレッシング、状態遷移マシン、知的連想マシン、オートマトンへの応用（言語理論・オートマトン理論）、大規模なデータベースから有用なルールあるいはパターンを高速に発見するデータマイニングなどを例として挙げることができる。応用範囲はこれらに限られるものではない。

請 求 の 範 囲

1. 検索対象のビットストリングを複数ビットの部分対象ビットストリングに分けて、予め登録された複数の登録ビットパターンと多段階で照合する過程を有
- 5 するビットストリングの照合方法であって、
前記多段階に含まれる1つの照合段階である現段階は、
前記検索対象のビットストリングから、前記現段階の前記部分対象ビットストリングを選択し、前記現段階の部分対象ビットストリングが取り得る全ての値と照合する全照合工程と、
- 10 前記複数の登録ビットパターンの各々の部分登録ビットパターンを示すためのパターンテーブルであって、前記現段階に対して先行する段階から得られる照合継続情報により決まる、前記現段階のパターンテーブルを、前記全照合工程と前後して、または並列に、メモリよりロードするパターンロード工程と、
前記全照合工程の結果と前記現段階のパターンテーブルとにより、前記現段階
- 15 の部分対象ビットストリングに一致する前記現段階の前記部分登録ビットパターンの有無を少なくとも示す照合結果を得る判定工程と、
前記照合結果により、前記現段階のパターンテーブルに対応するアドレステーブルから、前記現段階の次の段階の前記パターンテーブルのアドレスを含む前記照合継続情報を出力する出力工程とを有する照合方法。
- 20
2. 前記パターンテーブルは、前記部分登録ビットパターンの有効無効を示すビットフラグからなるマスクデータである、請求項1の照合方法。
3. 前記出力工程では、前記一致する現段階の部分登録ビットパターンに続く
- 25 前記次の段階の前記パターンテーブルのアドレスを含む前記照合継続情報を出力する、請求項1の照合方法。

4. 前記判定工程では、前記部分対象ビットストリングに値が最も近い最大または最小の前記現段階の部分登録ビットパターンの有無を含む前記照合結果が得られ、

5 前記出力工程では、前記最大または最小の現段階の部分登録ビットパターンに続く前記次の段階の範囲検索のパターンテーブルのアドレスを含む前記照合継続情報を出力する、請求項1の照合方法。

5. 前記一致する現段階の部分登録ビットパターンがあり、さらに、前記最大または最小の現段階の部分登録ビットパターンがあるときに、前記範囲検索のパターンテーブルのアドレスを候補アドレスとして記憶する候補アドレス記憶工程を有し、

前記一致する現段階の部分登録ビットパターンおよび前記最大または最小の現段階の部分登録ビットパターンがないときには、前記出力工程において、前記候補アドレスを含む前記照合継続情報を出力する、請求項4の照合方法。

15

6. 前記範囲検索のパターンテーブルのアドレスを含む前記照合継続情報が与えられたときは、前記全照合工程および判定工程をバイパスする工程を有し、

前記出力工程では、前記範囲検索のパターンテーブルに対応する前記アドレステーブルから、そのパターンテーブルに示された最大または最小の部分登録ビットパターンに続く次の段階のパターンテーブルのアドレスを含む前記照合継続情報を出力する、請求項4の照合方法。

20

7. 前記パターンテーブルは、そのパターンテーブルに示された最大または最小の部分登録ビットパターンに対応して決まる最大または最小の前記登録ビットパターンを示すバイパスデータを有し、

25

前記範囲検索のパターンテーブルのアドレスを含む前記照合継続情報が与えられたときは、前記全照合工程、前記パターンロード工程および前記判定工程をバイパスする工程を有し、

前記出力工程では、前記範囲検索のパターンテーブルに対応する前記バイパス

データを含む最終照合情報を出力して前記検索対象のビットストリングの照合を終了する、請求項 4 の照合方法。

8. 前記パターンテーブルおよび前記アドレステーブルを更新することにより、
5 前記登録ビットパターンを追加または削除する更新工程を有する、請求項 1 の照合方法。

9. 前記全照合工程と前記パターンロード工程とを並列に行う、請求項 1 の照合方法。

10

10. 複数の前記照合段階がパイプライン方式で実行される、請求項 1 の照合方法。

11. それぞれの前記照合段階の前記パターンテーブルおよび前記アドレス
15 テーブルを含む照合テーブルを更新することにより、前記登録ビットパターンを追加または削除する更新工程を有し、前記照合段階と並列にパイプライン方式で実行される、請求項 10 の照合方法。

12. 前記照合テーブルは、前記パターンテーブルに示された最大また最小の
20 部分登録ビットパターンに対応して決まる最大または最小の前記登録ビットパターンを示すバイパスデータを有し、

前記更新工程においては、上位の前記照合段階の前記照合テーブルから下位の前記照合段階の前記照合テーブルを更新し、その際、前記バイパスデータの更新が発生する可能性のある前記照合段階の前記照合テーブルの更新は、前記下位の
25 前記照合段階の前記照合テーブルの更新を待ってから行う、請求項 11 の照合方法。

1 3. 前記照合テーブルは、前記パターンテーブルに示された最大また最小の部分登録ビットパターンに対応して決まる最大または最小の前記登録ビットパターンを示すバイパスデータを有し、

5 前記更新工程においては、上位の前記照合段階の前記照合テーブルから下位の前記照合段階の前記照合テーブルを更新する工程と、前記下位の照合テーブルの更新により前記上位の照合テーブルの前記バイパスデータの更新が発生した場合は、双方向リンクにより前記上位の照合テーブルの更新を行う工程とを有する、請求項 1 1 の照合方法。

10 1 4. 複数の前記照合段階の間では、前記照合継続情報を含むデータがパケット化されて伝達される、請求項 1 の照合方法。

1 5. 請求項 4 に記載の照合段階を複数備えた分類過程を有し、前記登録ビットパターンは複数の分類結果を含む範囲を示している分類方法。

15

1 6. 複数の前記分類過程と、

前記複数の分類過程により得られる複数の前記範囲に含まれる前記複数の分類結果の論理積を演算し、最終の分類結果を得る論理演算過程とを有する、請求項 1 5 の分類方法。

20

1 7. 前記論理演算過程では、前記複数の分類結果の論理積を行列演算し、前記最終の分類結果を得る、請求項 1 6 の分類方法。

25 1 8. 請求項 1 5 に記載の分類方法により、IP アドレス、ポート番号およびプロトコルの少なくともいずれかを含むパケットを管理するためのビットストリングを前記検索対象のビットストリングとし、前記検索対象のビットストリングが属する前記範囲を照合する工程と、

前記検索対象のビットストリングが属する前記範囲の前記分類結果に基づき前記検索対象のビットストリングを備えた前記パケットを管理する工程とを有する

データ管理方法。

19. 請求項1に記載の照合方法により、前記検索対象のビットストリームを照合する工程と、

- 5 その照合する工程の照合結果に基づき前記検索対象のビットストリングを備えたデータを管理する工程とを有するデータ管理方法。

20. 請求項1に記載の照合方法により、状態遷移の評価元となるデータを前記検索対象のビットストリングとし、複数の状態遷移条件を示すデータを前記複数の登録ビットパターンとして照合する工程と、

- 10 前記照合する工程の照合結果によりデータ処理回路の状態を遷移する工程とを有するデータ処理装置の制御方法。

21. 検索対象のビットストリングを複数ビットの部分対象ビットストリングに分けて、予め登録された複数の登録ビットパターンと多段階で照合するために、前記多段階に含まれる少なくとも1つの段階を実行する照合装置であって、

- 15 前記検索対象のビットストリングから、前記多段階の1つの現段階の前記部分対象ビットストリングを選択し、その部分対象ビットストリングが取り得る全ての値と照合する全照合手段と、
- 20 前記複数の登録ビットパターンの部分登録ビットパターンを示すためのパターンテーブルであって、前記現段階に対して先行する段階から得られる照合継続情報により決まる、前記現段階のパターンテーブルをメモリよりロードするパターンロード手段と、

- 25 前記全照合手段の結果と前記現段階のパターンテーブルとにより、前記現段階の部分対象ビットストリングに一致する前記現段階の部分登録ビットパターンの有無を少なくとも示す照合結果を出力する判定手段と、

前記照合結果により、前記現段階のパターンテーブルに対応するアドレステーブルから、前記現段階の次の段階の前記パターンテーブルのアドレスを含む前記照合継続情報を出力する出力手段とを有する照合装置。

22. 前記パターンテーブルは、前記部分登録ビットパターンの有効無効を示すビットフラグからなるマスクデータである、請求項21の照合装置。

- 5 23. 前記出力手段は、前記一致する現段階の部分登録ビットパターンに続く前記次の段階の前記パターンテーブルのアドレスを含む前記照合継続情報を出力する、請求項21の照合装置。

- 10 24. 前記判定手段は、前記部分対象ビットストリングに値が最も近い最大または最小の前記現段階の部分登録ビットパターンの有無を含む前記照合結果を出力し、

前記出力手段は、前記最大または最小の現段階の部分登録ビットパターンに続く前記次の段階の範囲検索のパターンテーブルのアドレスを含む前記照合継続情報を出力する、請求項21の照合装置。

15

25. 前記出力手段は、さらに、前記一致する部分登録ビットパターンがあり、前記最大または最小の現段階の部分登録ビットパターンがあるときに、前記範囲検索のパターンテーブルのアドレスを候補アドレスとして出力し、さらに、

- 20 前記一致する現段階の部分登録ビットパターンおよび前記最大または最小の現段階の部分登録ビットパターンがないときに、前記候補アドレスを含む前記照合継続情報を出力する、請求項24の照合装置。

26. 前記出力手段は、前記範囲検索のパターンテーブルのアドレスを含む前記照合継続情報が与えられたときは、前記範囲検索のパターンテーブルに対応する前記アドレステーブルから、そのパターンテーブルに示された最大または最小の部分登録ビットパターンに続く次の段階のパターンテーブルのアドレスを含む前記照合継続情報を出力する、請求項24の照合装置。
- 25

27. 前記パターンテーブルは、そのパターンテーブルに示された最大また最小の部分登録ビットパターンに対応して決まる最大または最小の前記登録ビットパターンを示すバイパスデータを有し、

5 前記出力手段は、前記範囲検索のパターンテーブルのアドレスを含む前記照合継続情報が与えられたときは、前記範囲検索のパターンテーブルに対応する前記バイパスデータを最終照合情報として出力する、請求項24の照合装置。

10 28. 前記パターンテーブルおよび前記アドレステーブルを更新することにより、前記登録ビットパターンを追加または削除する更新手段を有する、請求項21の照合装置。

29. 前記全照合手段と、前記パターンロード手段とは並列に動作する、請求項21の照合装置。

15 30. 請求項21に記載の照合装置を複数有する分類装置。

31. 前記複数の照合装置はパイプラインで接続されている、請求項30の分類装置。

20 32. 前記複数の照合装置の間では、前記照合継続情報を含むデータがパケット化されて伝達される、請求項30の分類装置。

33. 前記メモリに対してデータを入出力するキャッシュ装置を有する、請求項30の分類装置。

25

34. 前記メモリには、前記パターンテーブルおよび前記アドレステーブルを含む照合テーブルが記憶されており、前記キャッシュ装置のキャッシュラインのサイズが前記照合テーブルのサイズと同一である、請求項33の分類装置。

35. 最初の照合テーブルを指定する検索種別情報を受信する手段を有し、
前記キャッシュ装置は、それぞれの検索種別および前記照合段階の単位で前記
キャッシュラインを割り当て、前記検索種別および照合段階の単位で前記キャッ
シュラインを管理する管理手段を備えている、請求項34の分類装置。

5

36. 最初の照合テーブルを指定する検索種別情報を受信する手段を有し、
前記メモリには、それぞれの検索種別および前記照合段階の単位で割り当てら
れた個別アドレス領域に前記照合テーブルが記憶されており、

前記キャッシュ装置の前記キャッシュラインは、前記個別アドレス領域単位で
10 割り当てられている、請求項34の分類装置。

37. 前記キャッシュ装置は、照合テーブルがオンキャッシュでなければ前記
照合装置にその旨を通知し、その照合装置の処理を中断して待ち行列に戻す、請
求項33の分類装置。

15

38. 当該分類装置の分類結果を記憶した履歴キャッシュ装置であって、前記
検索対象ビットストリングを前記照合装置に供給する前に、当該分類装置の分類
結果と前記検索対象のビットストリングとを比較する履歴キャッシュ装置を有す
る、請求項30の分類装置。

20

39. 請求項30に記載の分類装置を有し、前記登録ビットパターンは複数の
分類結果を含む範囲を示しており、前記検索対象のビットストリングが属する分
類結果を出力する検索装置。

25 40. 複数の前記分類装置と、

前記複数の分類装置により得られる複数の前記範囲に含まれる複数の分類結果
の論理積を演算し、最終の分類結果を得る論理演算装置を有する、請求項39の
検索装置。

4 1. 前記論理演算装置は、前記複数の分類結果の論理積を行列演算し、最終の分類結果を得る、請求項 4 0 の検索装置。

4 2. 前記メモリに対してデータを入出力するキャッシュ装置を有する、請求項 3 9 の検索装置。

4 3. 当該検索装置の検索結果を記憶した履歴キャッシュ装置であって、前記検索対象ビットストリングを前記分類装置に供給する前に、当該検索装置の検索結果と前記検索対象のビットストリングとを比較する履歴キャッシュを有する、請求項 3 9 の検索装置。

4 4. 請求項 4 0 に記載の検索装置を有し、
前記検索装置に、IP アドレス、ポート番号およびプロトコルの少なくともいずれかを含むパケットを管理するためのビットストリングを前記検索対象のビットストリングとして供給する手段と、

前記検索装置から出力された分類結果に基づき前記検索対象のビットストリングを備えた前記パケットを管理する手段とを有するデータ管理装置。

4 5. 請求項 3 0 に記載の分類装置と、
前記分類装置の出力に基づき前記検索対象のビットストリングを備えたデータを管理する手段とを有するデータ管理装置。

4 6. 請求項 3 0 に記載の分類装置を有し、前記複数の登録ビットパターンは、複数の状態遷移条件を示すデータであり、
前記分類装置に対し、状態遷移の評価元となるデータを前記検索対象のビットストリングとして供給する検索対象供給手段と、
前記分類装置の出力により状態が遷移するデータ処理回路とを有するデータ処理装置。

47. 前記メモリには、前記パターンテーブルおよび前記アドレステーブルを含む照合テーブルが記憶されており、

前記分類装置の出力または前記検索対象供給手段により、最初の前記照合装置に供給される前記照合テーブルが指定される、請求項46のデータ処理装置。

5

48. 前記メモリには、前記パターンテーブルおよび前記アドレステーブルを含む照合テーブルが記憶されており、

前記データ処理回路により前記メモリの照合テーブルが書き換えられる、請求項47のデータ処理装置。

10

49. 状態遷移の評価元となる現状態の検索対象のビットストリングを、予め登録された複数の状態遷移条件を示す登録ビットパターンと照合する照合装置と、

前記照合装置に対し、前記検索対象のビットストリングを供給する検索対象供給手段と、

15 前記照合装置の出力により状態が遷移するデータ処理回路とを有するデータ処理装置であって、

前記現状態の検索対象のビットストリングが取り得る全ての値と照合する全照合手段と、

20 前記複数の登録ビットパターンを示すためのパターンテーブルであって、先行する状態から得られる照合継続情報により決まる、現状態のパターンテーブルをメモリよりロードするパターンロード手段と、

前記全照合手段の結果と前記現状態のパターンテーブルとにより、前記現状態の検索対象ビットストリングに一致する前記現状態の登録ビットパターンの有無を少なくとも示す照合結果を出力する判定手段と、

25 前記照合結果により、前記現状態のパターンテーブルに対応するアドレステーブルから、前記現状態の次の状態の前記パターンテーブルのアドレスを含む前記照合継続情報を出力する出力手段とを備えている、データ処理装置。

50. 前記メモリには、前記パターンテーブルおよび前記アドレステーブルを含む照合テーブルが記憶されており、

前記検索対象供給手段により、前記照合装置に供給される前記照合テーブルが指定される、請求項49のデータ処理装置。

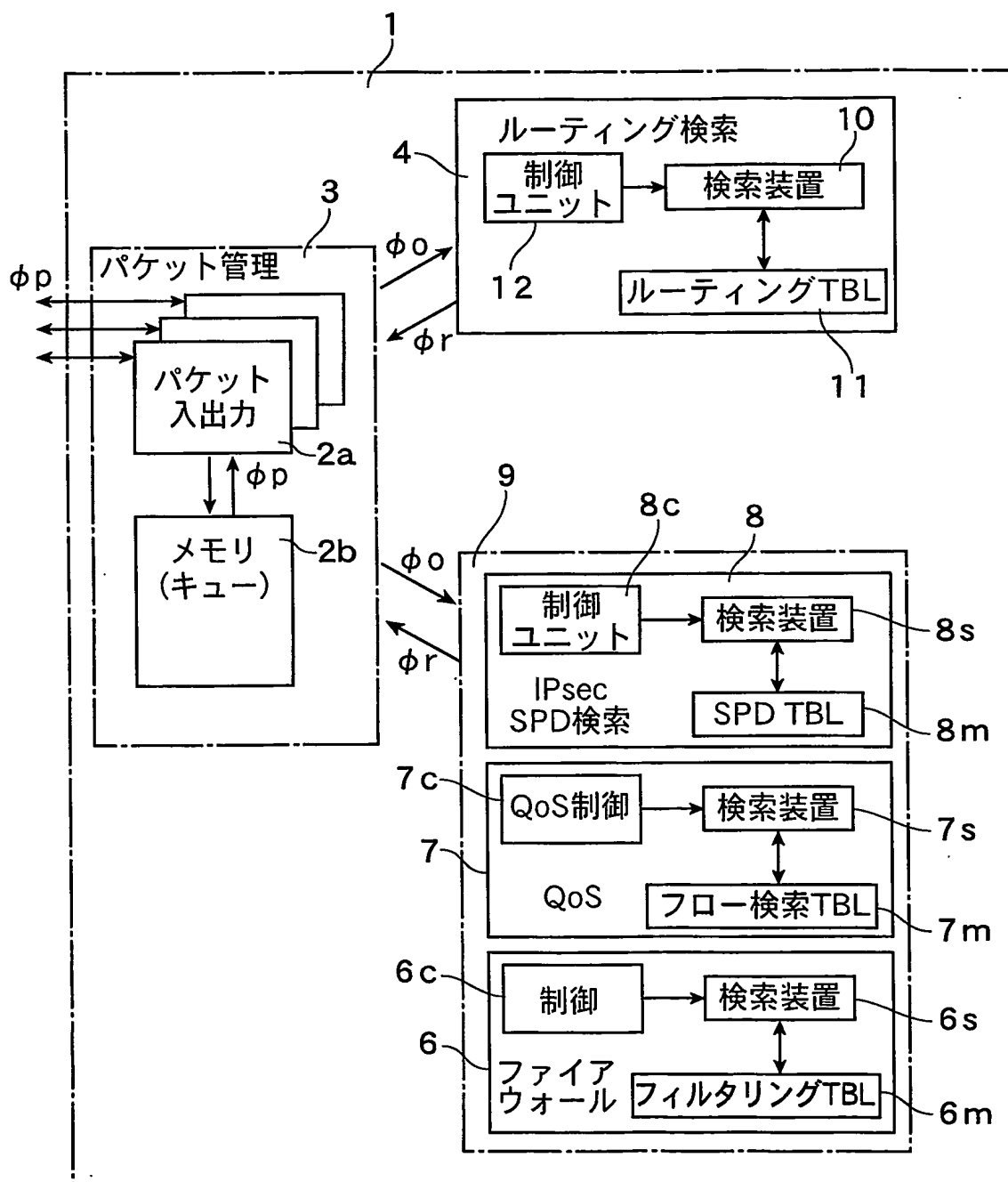
5

51. 前記メモリには、前記パターンテーブルおよび前記アドレステーブルを含む照合テーブルが記憶されており、

前記データ処理回路により前記メモリの照合テーブルが書き換えられる、請求項49のデータ処理装置。

1 / 5 3

図 1



2 / 53

図 2

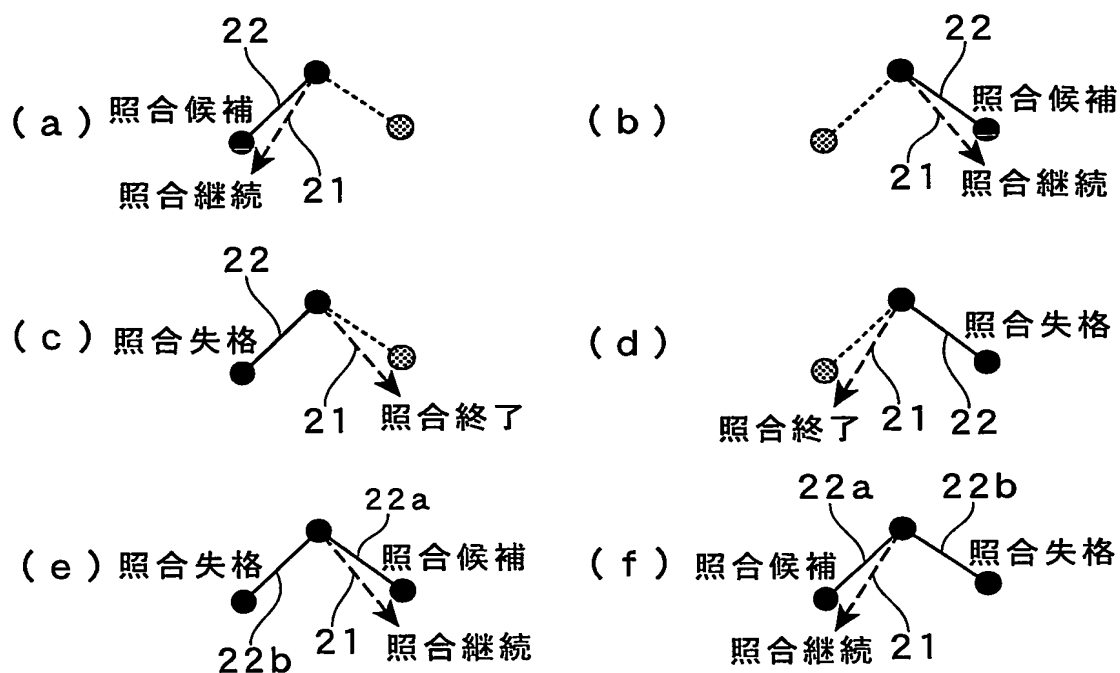
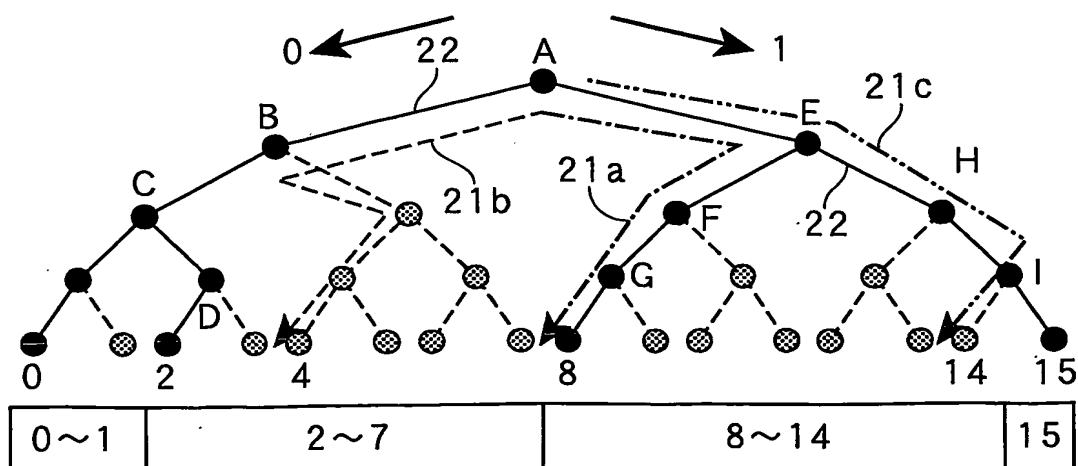


図 3



3 / 5 3

图 4

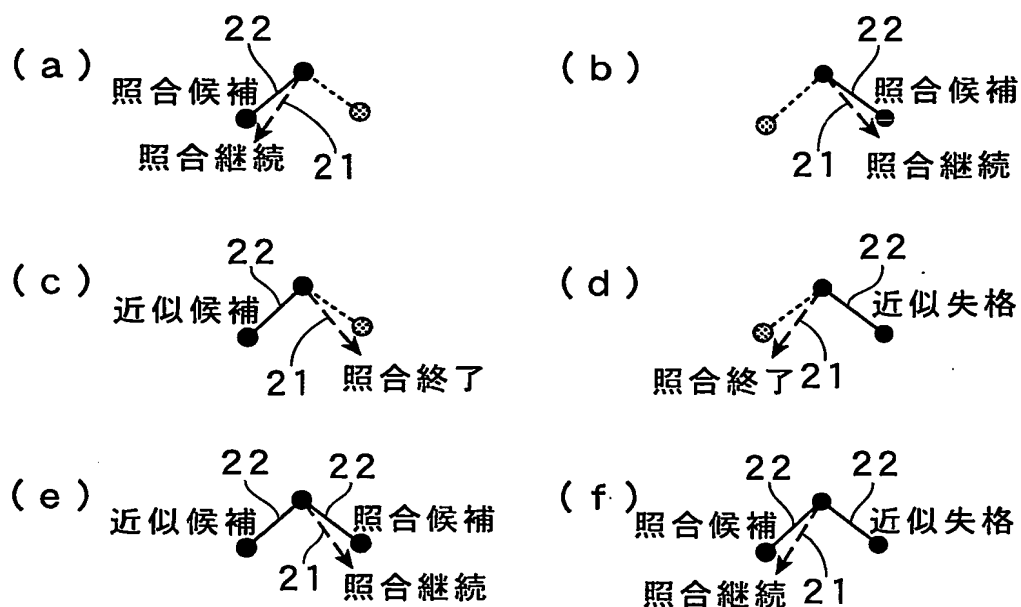
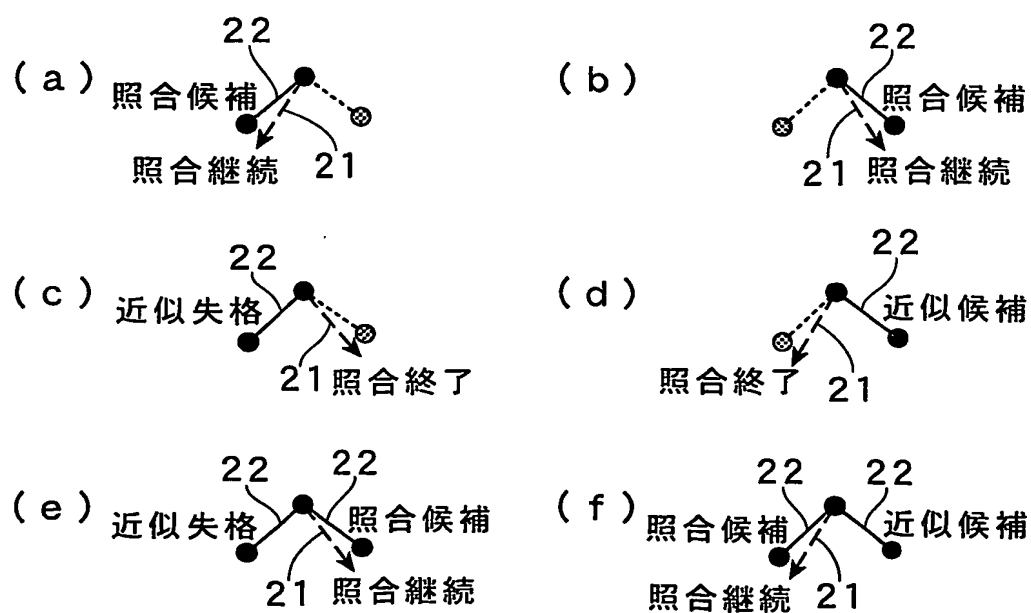


图 5



4 / 5 3

図 6

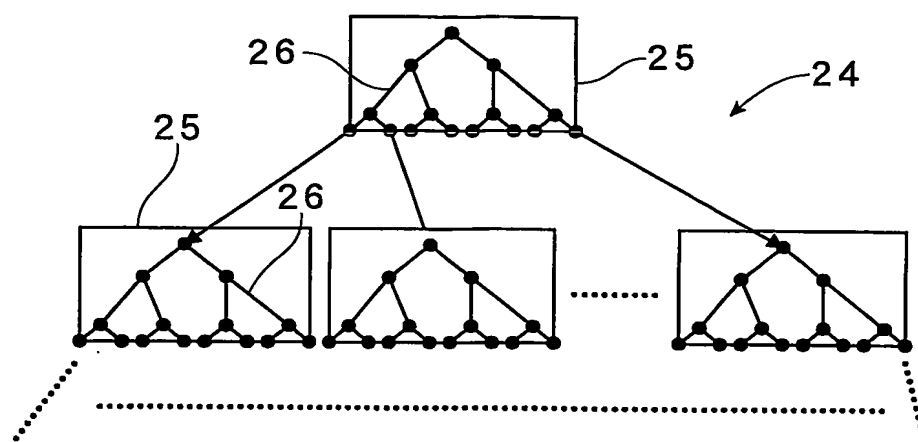
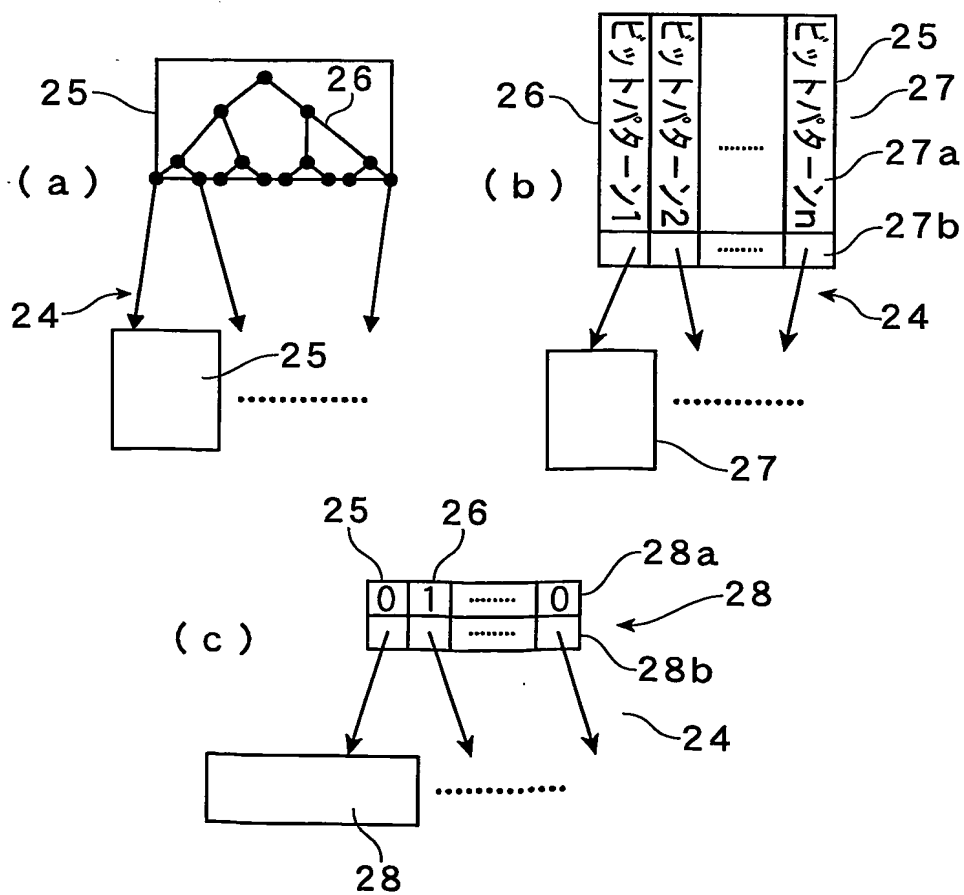


図 7



5 / 5 3

图 8

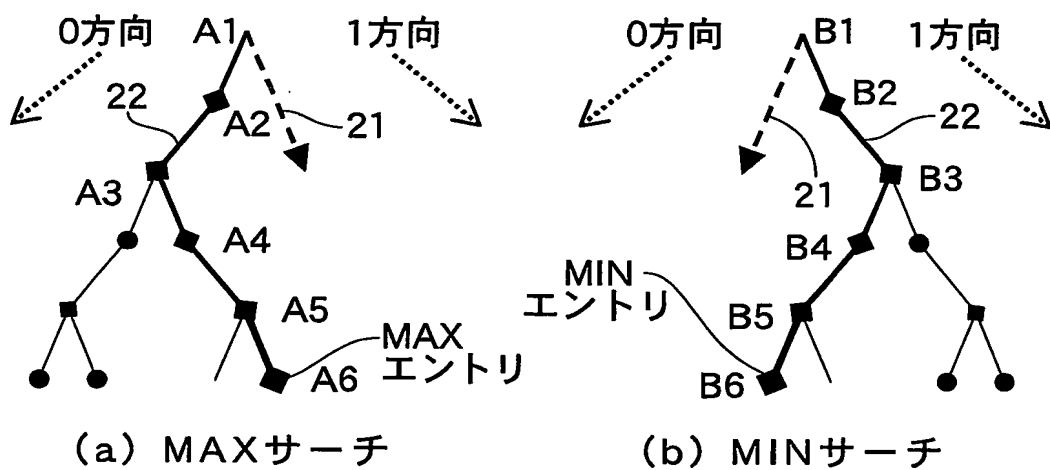
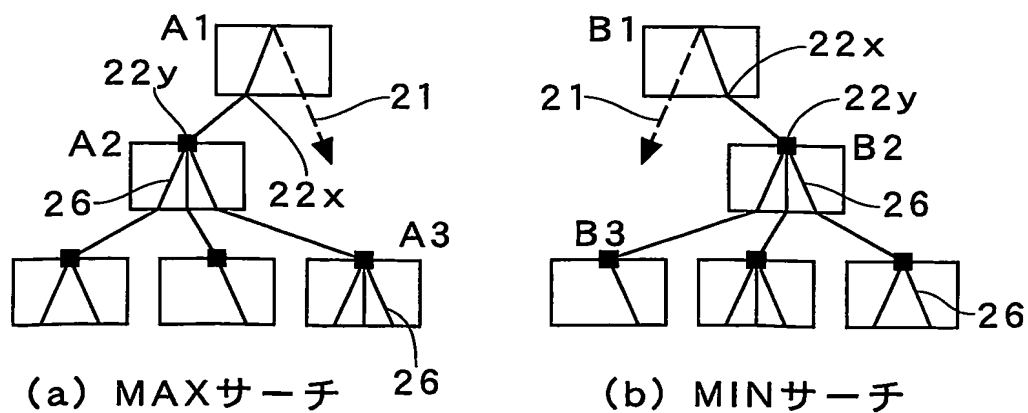
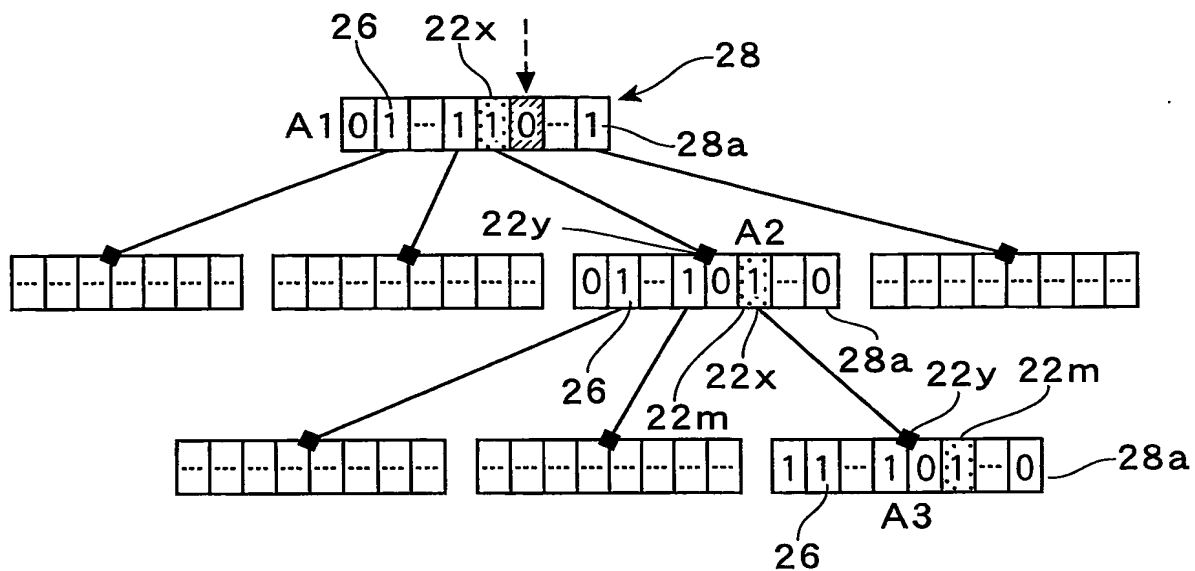


图 9



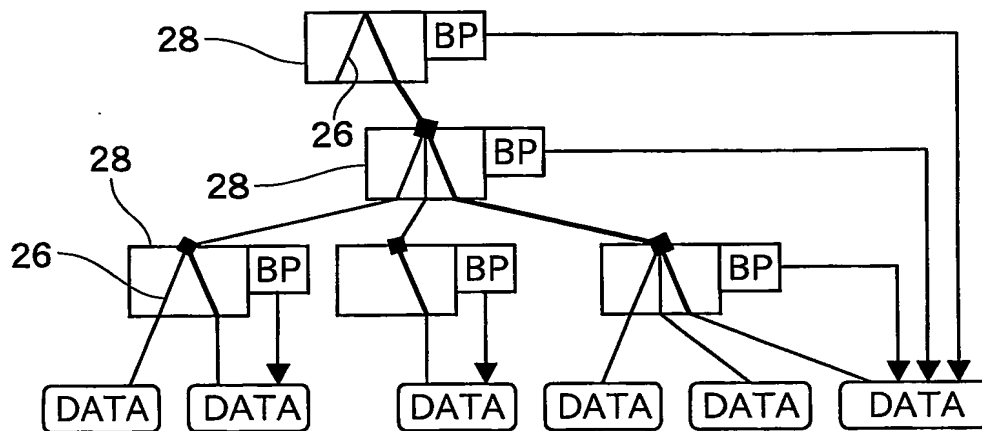
6 / 5 3

図 10

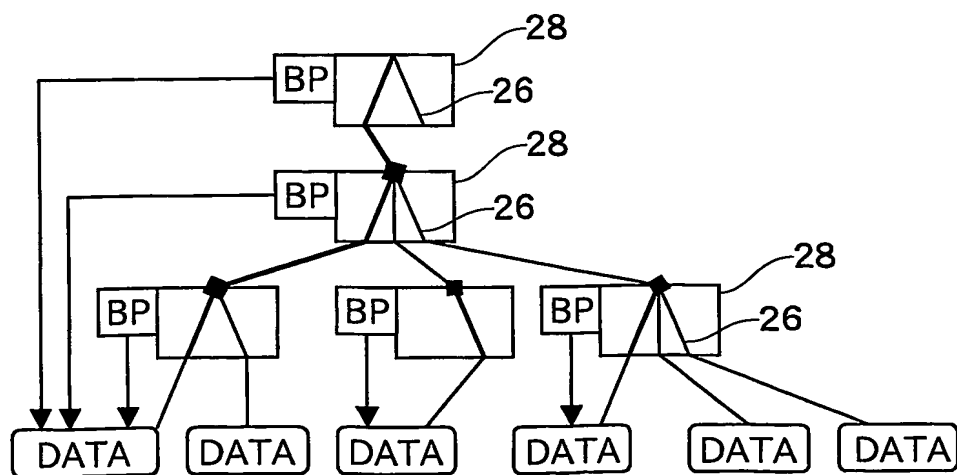


7 / 5 3

図 1 1



(a) MAXバイパス



(b) MINバイパス

8 / 5 3

図 1 2

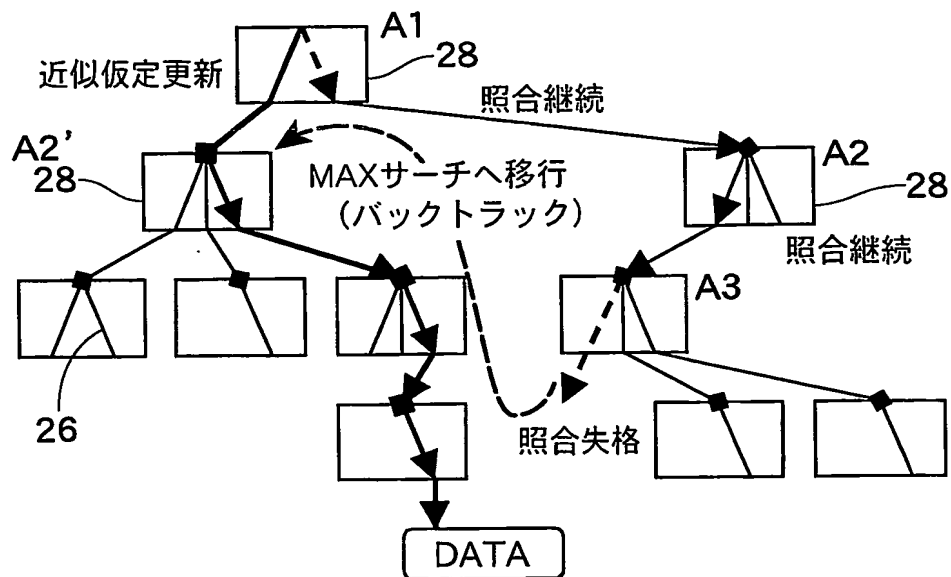
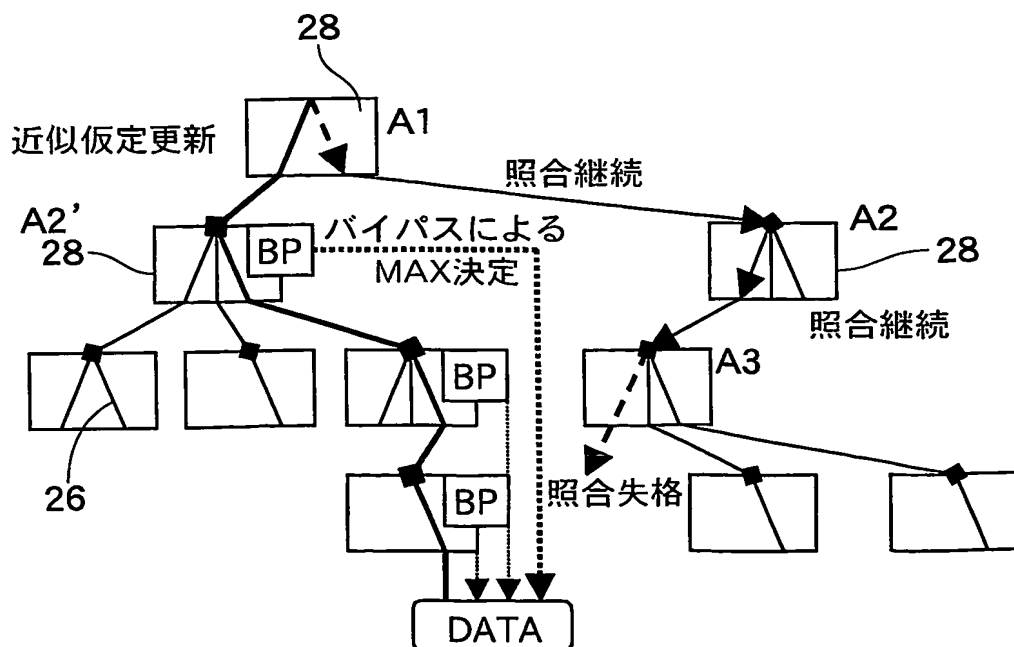
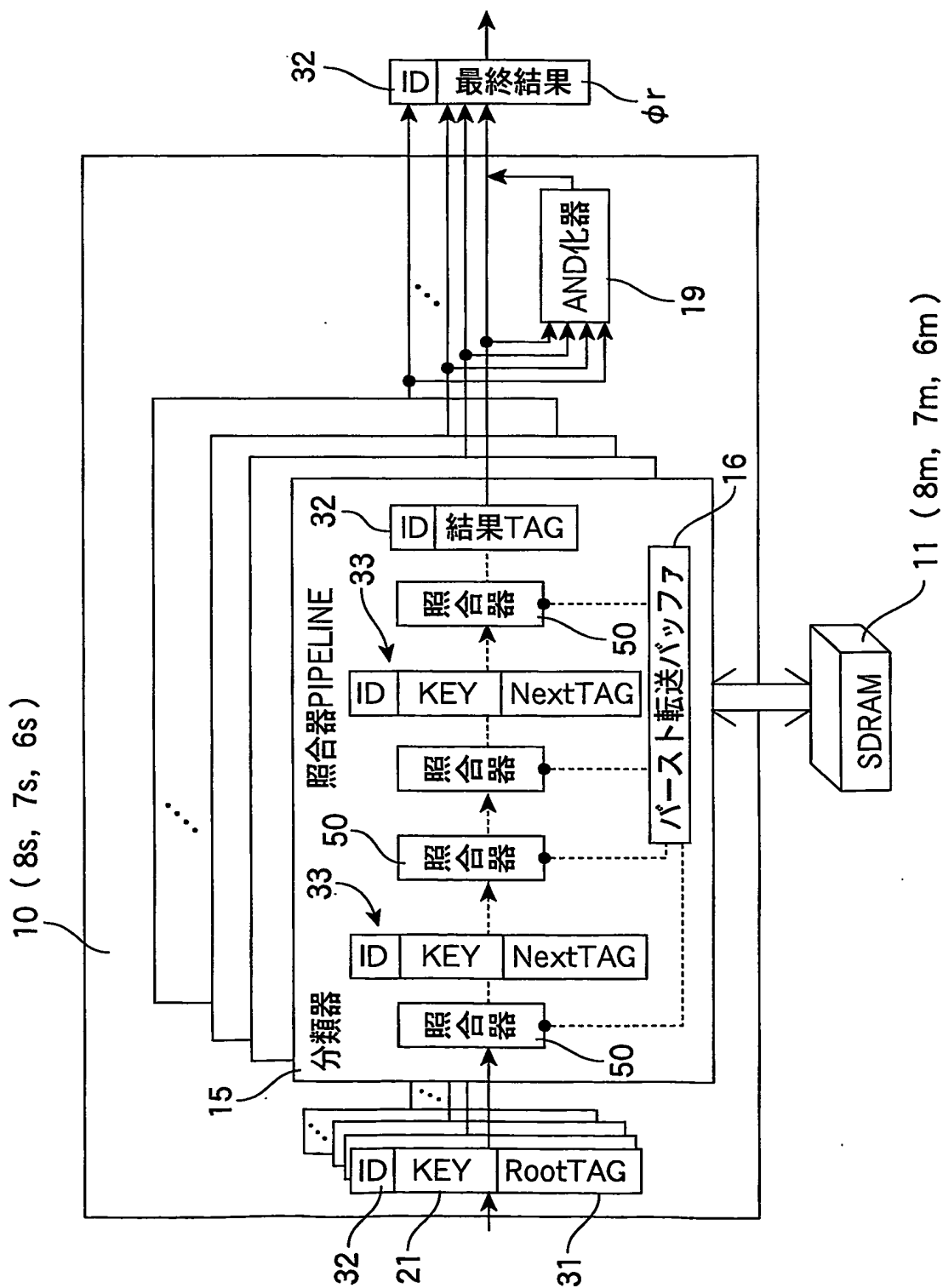


図 1 3



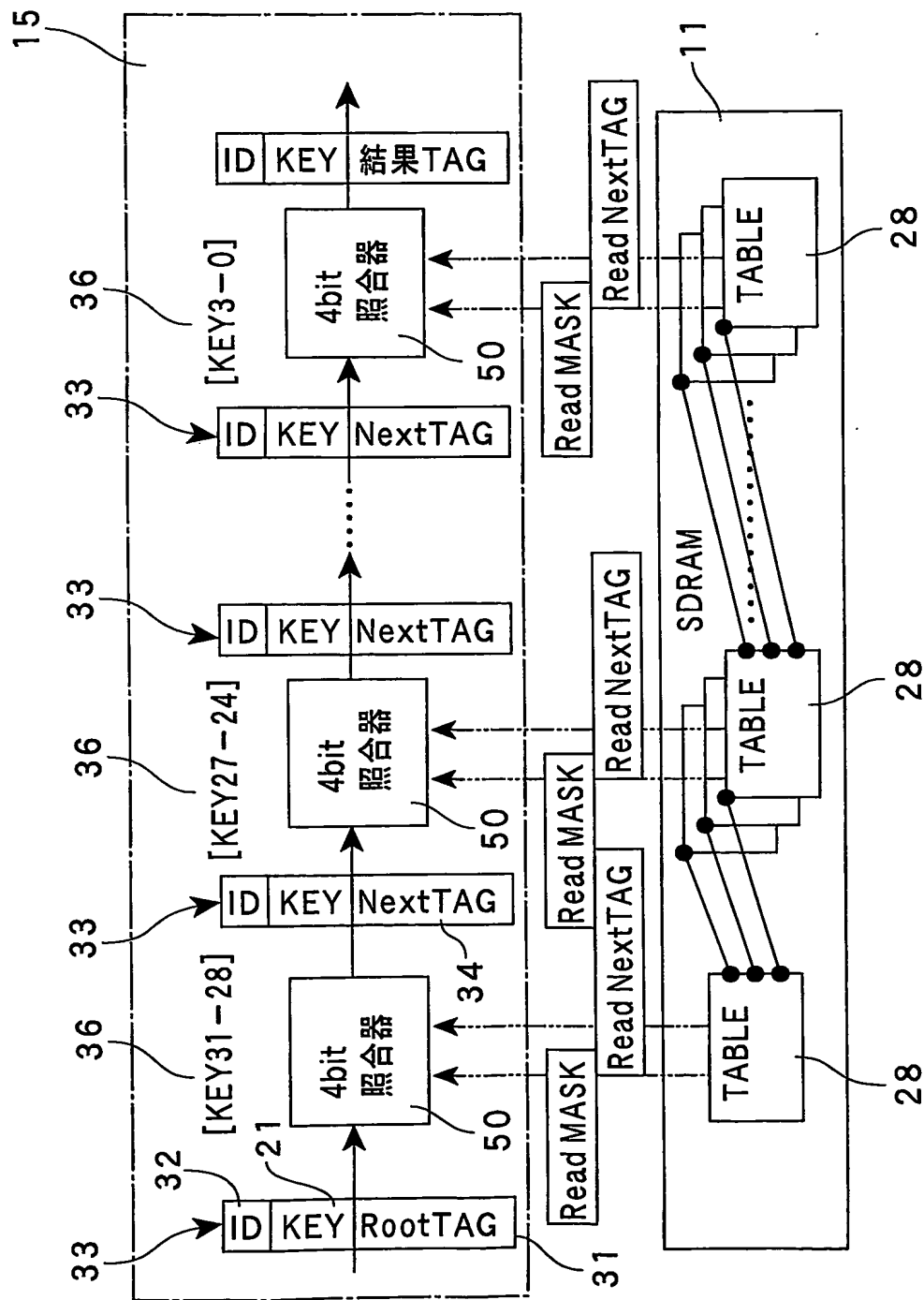
9 / 5 3

図14



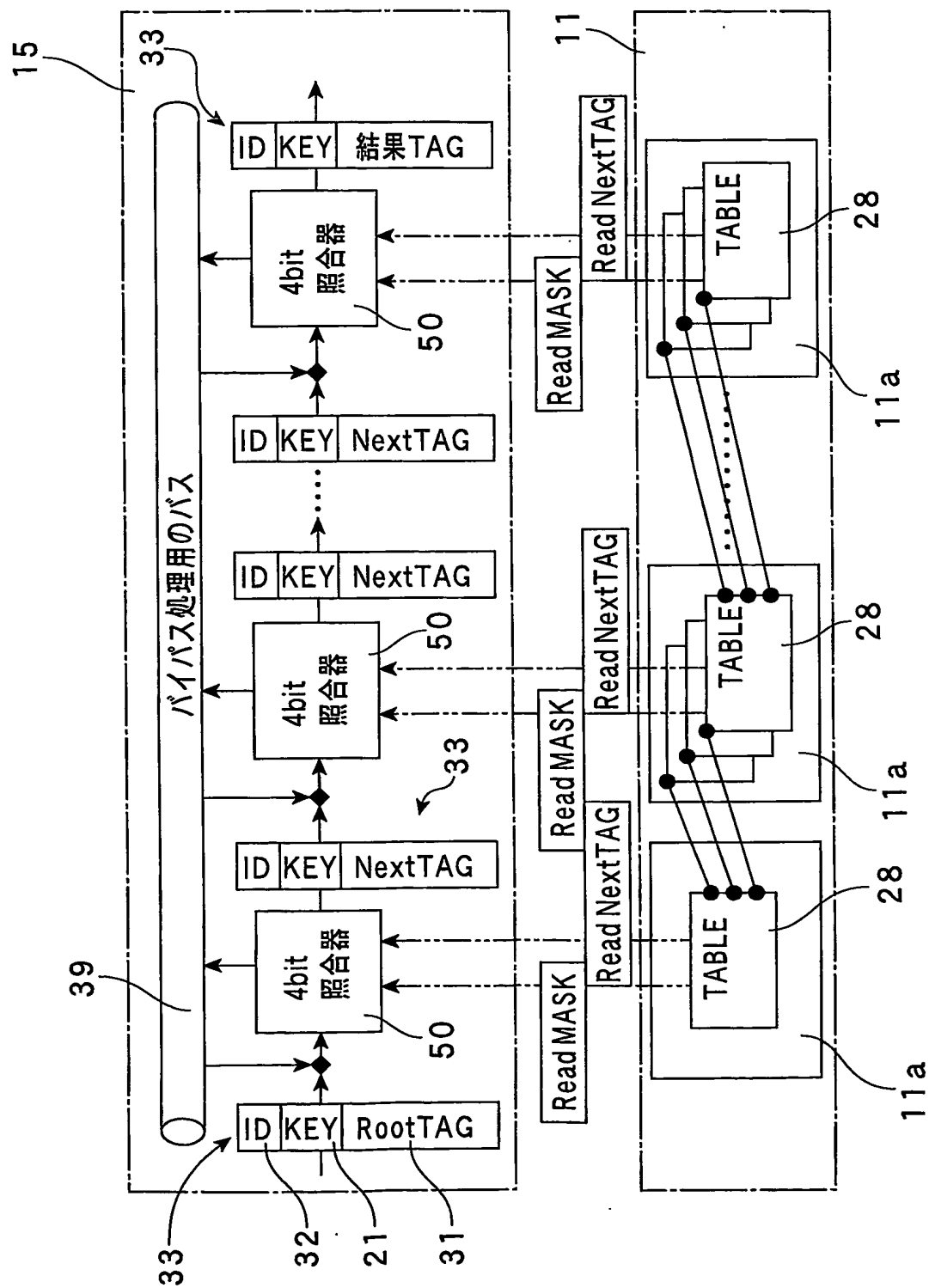
10 / 53

图 15



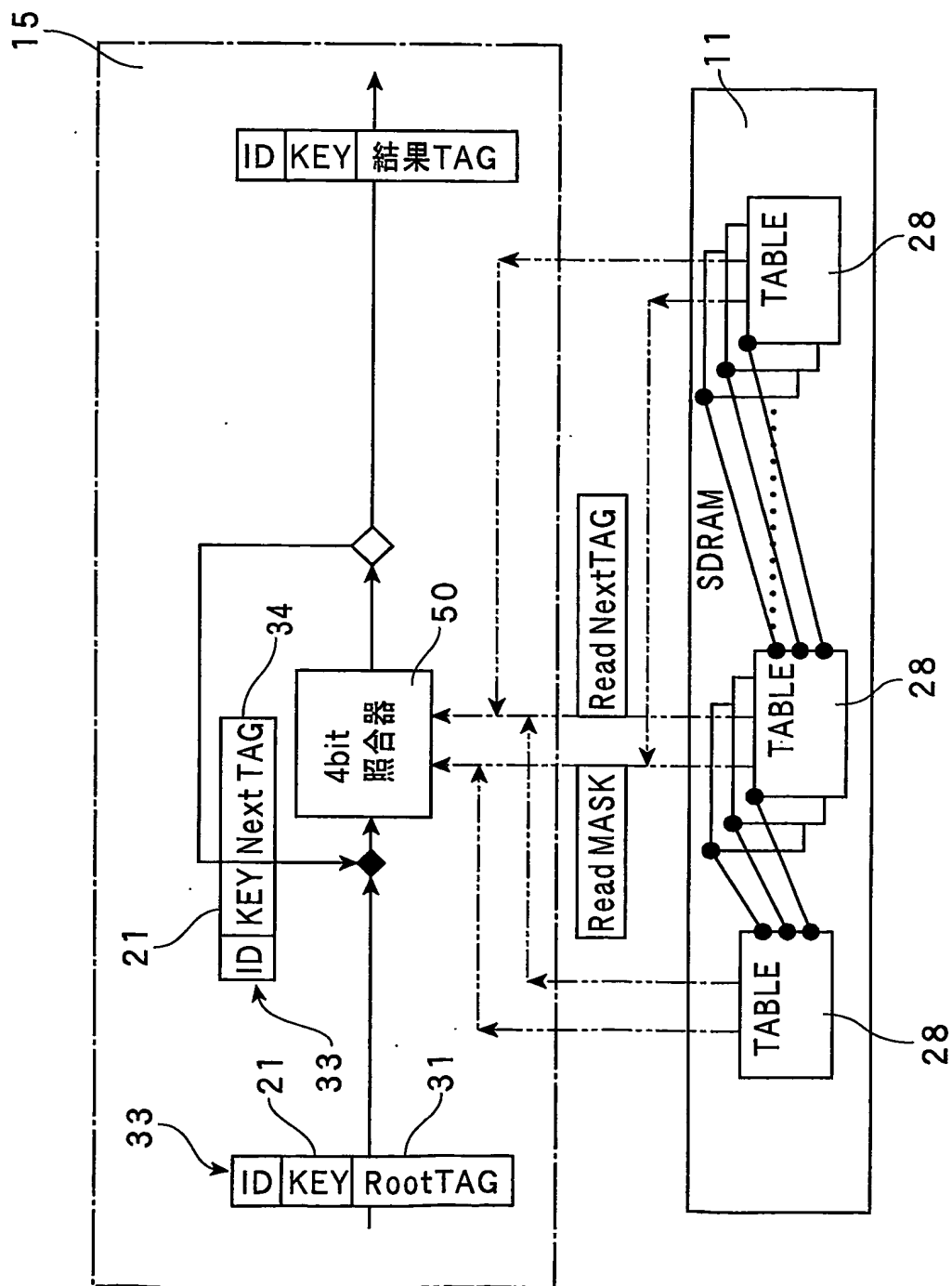
11 / 53

図16

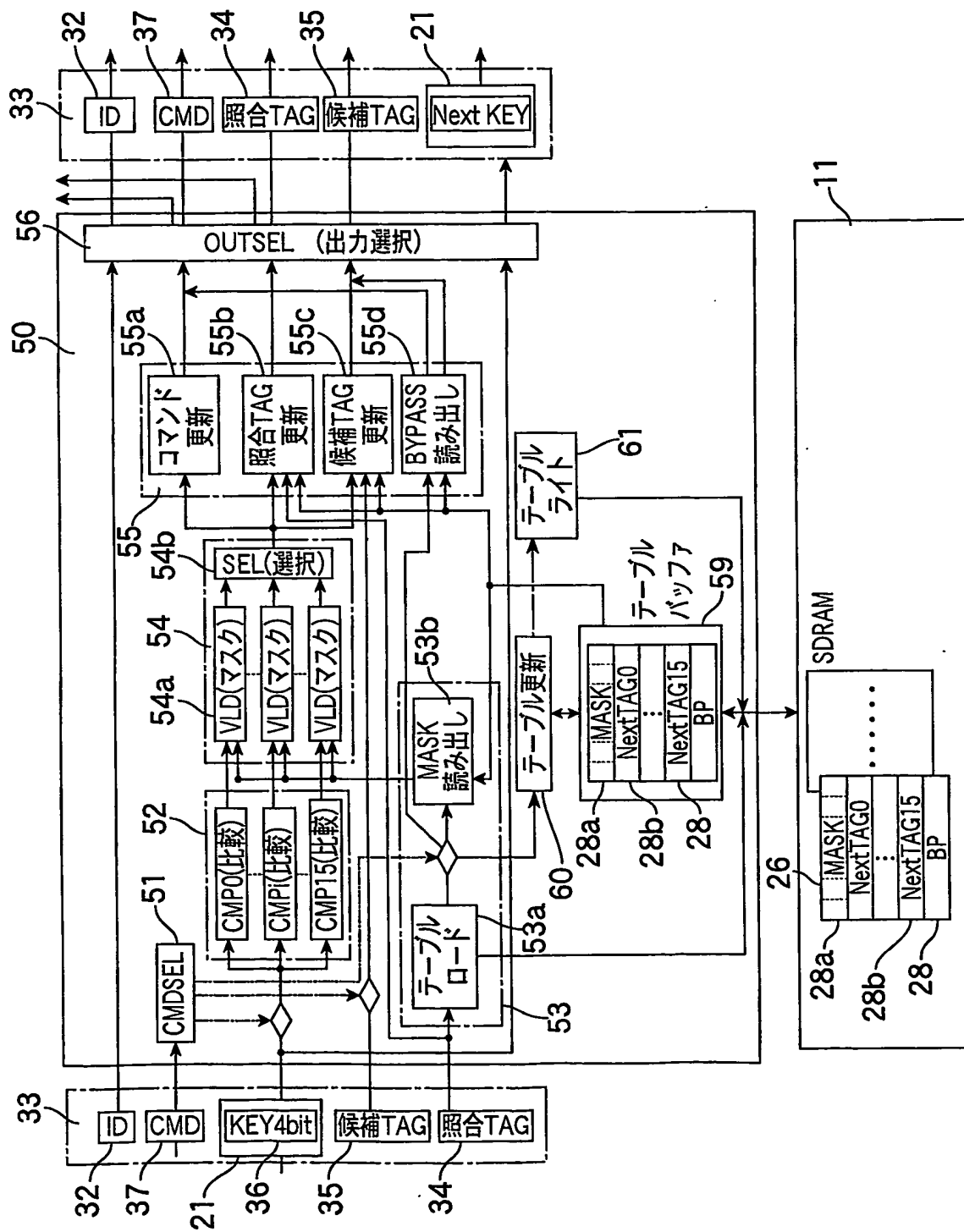


1 2 / 5 3

図 17

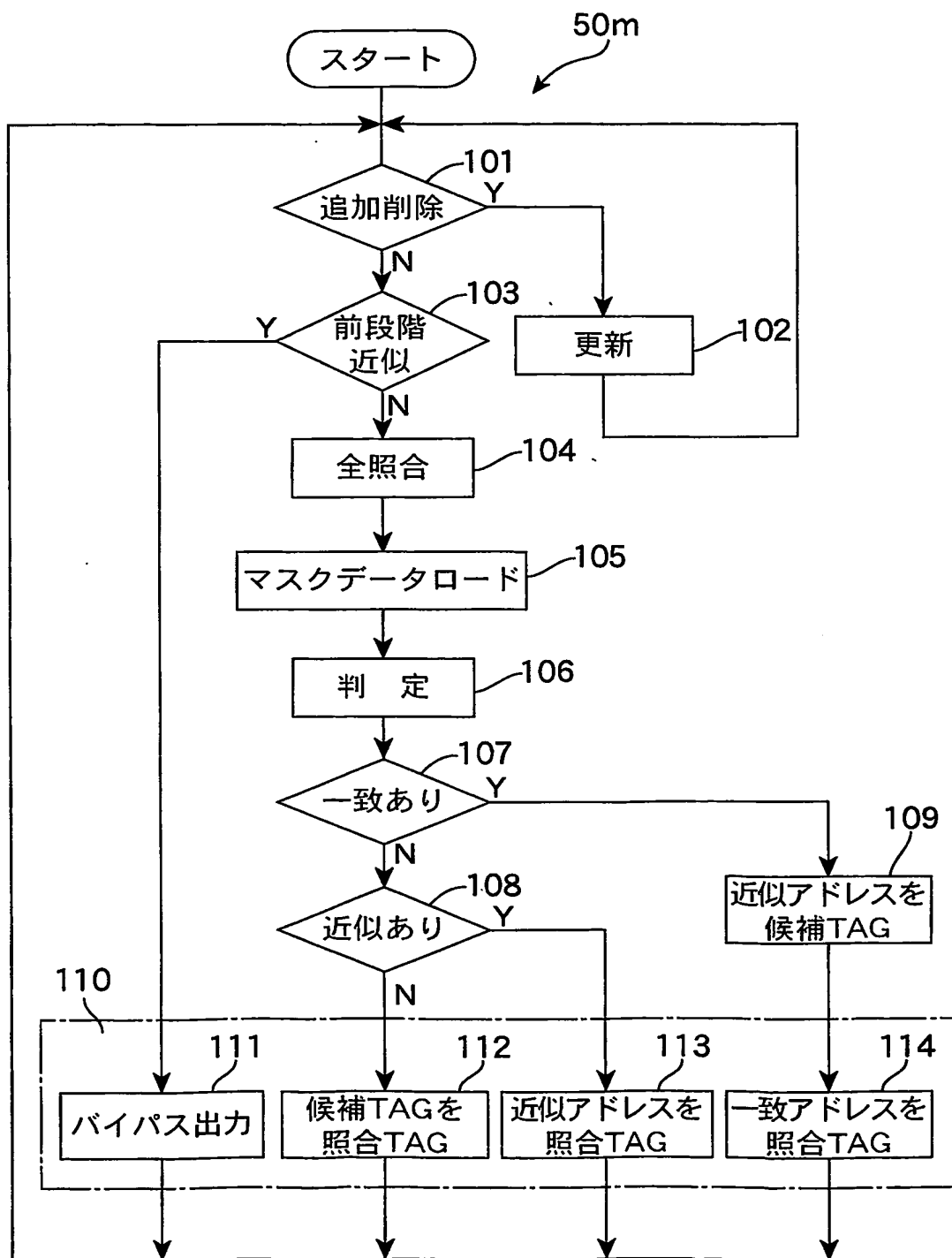


81X



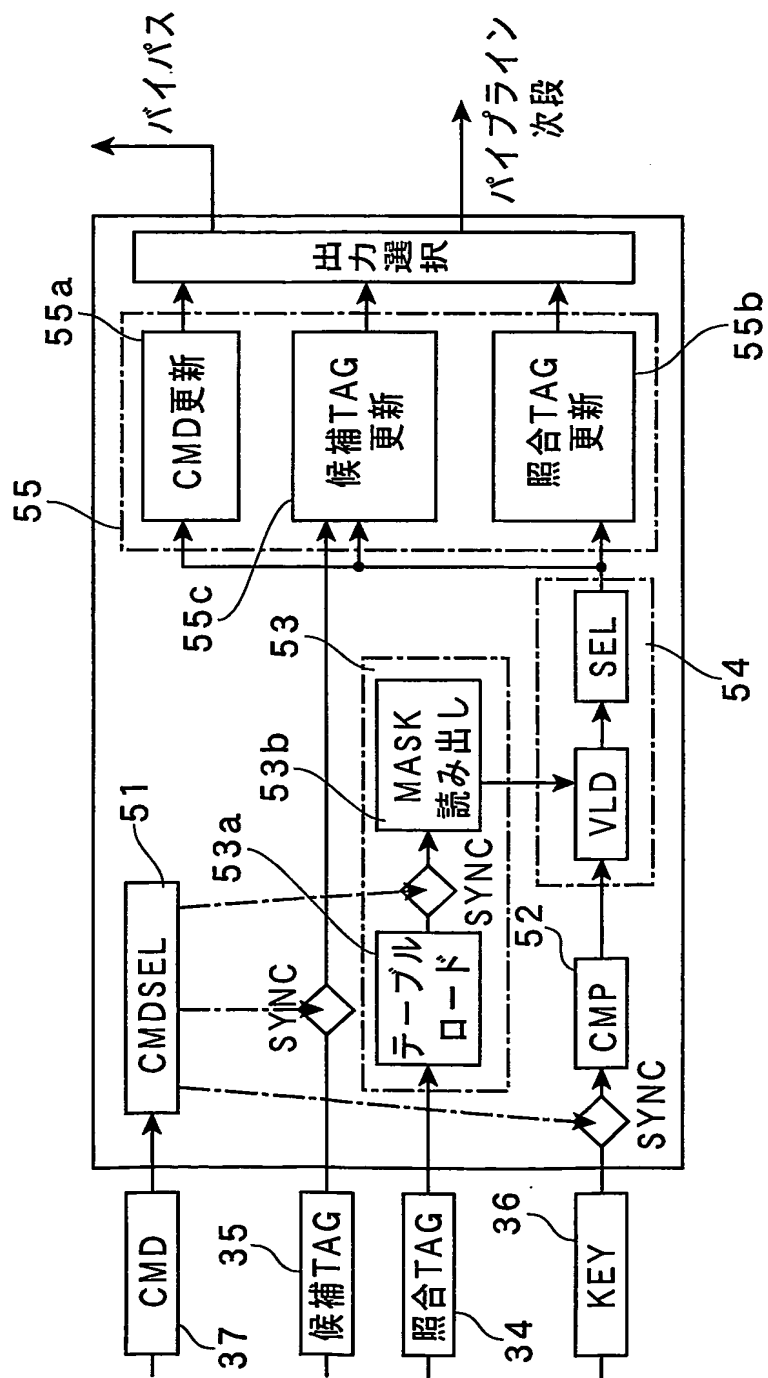
1 4 / 5 3

図 1 9



1 5 / 5 3

図 20



1 6 / 5 3

図 2 1

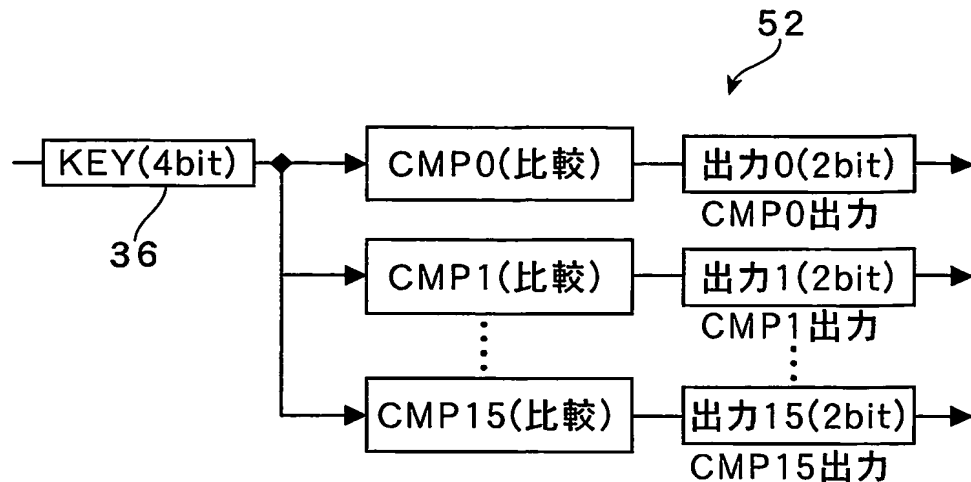


図 2 2

比較器 CMP i の演算論理

CMP i 入力 x	出力
$x < i$	10 (「<」)
$x = i$	01 (「=」)
$x > i$	11 (「>」)
i = 0 ~ 15 (固定値)	
x = 0 ~ 15 (4 bit入力値)	

図 2 3

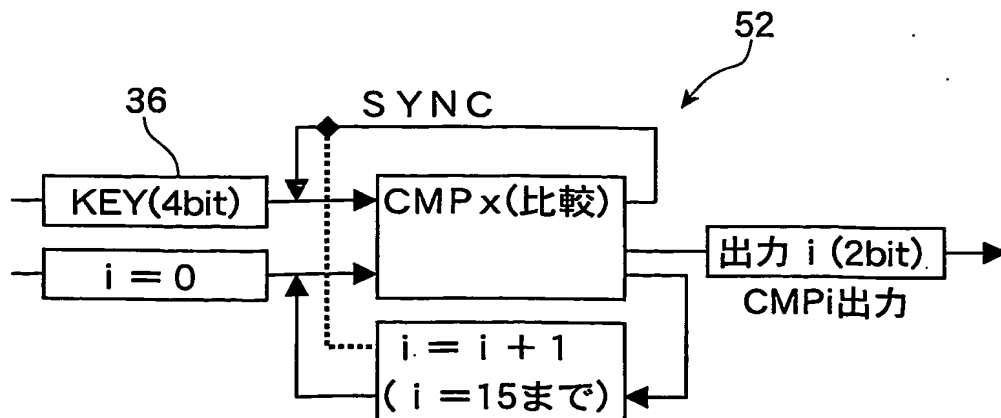


図 2 4

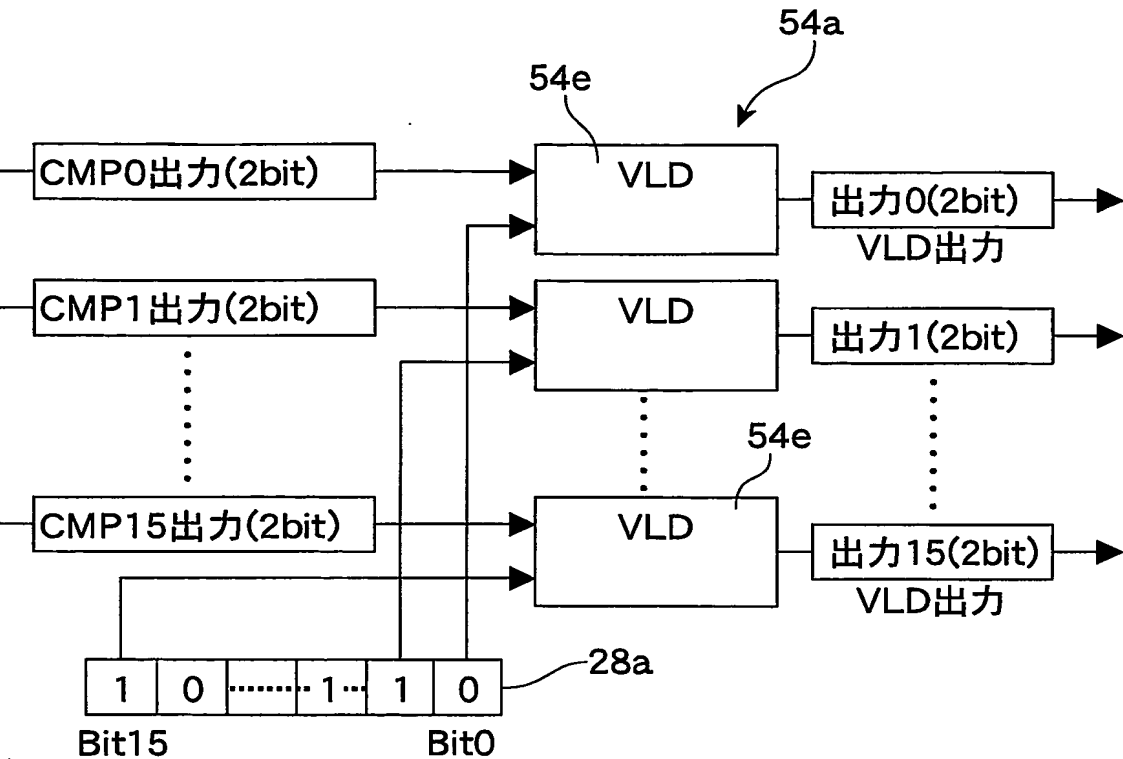


図 2 5

マスク器VLDの演算論理

CMPi出力	MASK値	VLD出力
10	0	00 (「X」)
01	0	00 (「X」)
11	0	00 (「X」)
10	1	10 (「<」)
01	1	01 (「=」)
11	1	11 (「>」)
MASK値によるAND演算		

18/53

図 2 6

完全一致検索の選択論理

V L D 出力	出力 G_f
00 ($\Gamma \times J$)	0
10 ($\Gamma < J$)	0
01 ($\Gamma = J$)	1
11 ($\Gamma > J$)	0

図 2 7

範囲検索の選択論理 (右倒れ一致型)

(a)

V L D 出力	中間 F_r	出力 G_r
00 ($\Gamma \times J$)	0	0
10 ($\Gamma < J$)	0	0
01 ($\Gamma = J$)	0	0
11 ($\Gamma > J$)	1	$G_r(i)$

(b)

$H_r(i-1)$	$F_r(i)$	$H_r(i)$
0	0	0
0	1	1
1	0	1
1	1	1

(c)

$H_r(i-1)$	$H_r(i)$	$G_r(i)$
0	0	0
0	1	1
1	0	0
1	1	0

19/53

図 28

範囲検索の選択論理 (左倒れ一致)

(a)

V L D 出力	中間 F_I	出力 G_I
00 ($\Gamma \times J$)	0	0
10 ($\Gamma < J$)	1	$G_I(i)$
01 ($\Gamma = J$)	0	0
11 ($\Gamma > J$)	0	0

(b)

$H_I(i+1)$	$F_I(i)$	$H_I(i)$
0	0	0
0	1	1
1	0	1
1	1	1

(c)

$H_I(i+1)$	$H_I(i)$	$G_I(i)$
0	0	0
0	1	1
1	0	0
1	1	0

20 / 53

図 2 9

エントリ0

エントリ15

(a) CMP出力

11	11	...	11	11	01	10	10	...	10	10
----	----	-----	----	----	----	----	----	-----	----	----

(b) マスク値 ×

0	1	...	0	1	1	1	0	...	0	1
---	---	-----	---	---	---	---	---	-----	---	---

(c) VLD出力 II

00	11	...	00	11	01	10	00	...	00	10
----	----	-----	----	----	----	----	----	-----	----	----

(c) VLD出力

00	11	...	00	11	01	10	00	...	00	10
----	----	-----	----	----	----	----	----	-----	----	----

(d) F_r 出力

0	1	...	0	1	0	0	0	...	0	0
---	---	-----	---	---	---	---	---	-----	---	---

(e) H_r 出力

1	1	...	1	1	0	0	0	...	0	0
---	---	-----	---	---	---	---	---	-----	---	---

(f) G_r 出力

0	0	...	0	1	0	0	0	...	0	0
---	---	-----	---	---	---	---	---	-----	---	---

(g) G_f 出力

0	0	...	0	0	1	0	0	...	0	0
---	---	-----	---	---	---	---	---	-----	---	---

図 3 0

エントリ0

エントリ15

(a) CMP出力

11	11	...	11	11	01	10	10	...	10	10
----	----	-----	----	----	----	----	----	-----	----	----

(b) マスク値 ×

1	0	...	0	1	1	1	0	...	1	0
---	---	-----	---	---	---	---	---	-----	---	---

(c) VLD出力 II

11	00	...	00	11	01	10	00	...	10	00
----	----	-----	----	----	----	----	----	-----	----	----

(c) VLD出力

11	00	...	00	11	01	10	00	...	10	00
----	----	-----	----	----	----	----	----	-----	----	----

(d) F_l 出力

0	0	...	0	0	0	1	0	...	1	0
---	---	-----	---	---	---	---	---	-----	---	---

(e) H_l 出力

0	0	...	0	0	0	1	1	...	1	1
---	---	-----	---	---	---	---	---	-----	---	---

(f) G_l 出力

0	0	...	0	0	0	1	0	...	0	0
---	---	-----	---	---	---	---	---	-----	---	---

(g) G_f 出力

0	0	...	0	0	0	1	0	...	0	0
---	---	-----	---	---	---	---	---	-----	---	---

21 / 53

図 3 1

37

種 別	コマン	説明
完全一致検索	FMATCHEXEC	照合継続
	FMATCHHIT	ヒット
	FMATCHFAIL	ミスヒット
範囲検索	MATCHEXEC	照合継続
	MATCHRDBP	バイパス読み出し
	MATCHFHIT	完全一致ヒット
	MATCHNHIT	近似ヒット
	MATCHFAIL	ミスヒット
登 録	ADD	上位→下位
	ADDNEW	新規テーブル
	ADDFIN	登録完了
	ADDFAIL	登録失敗
	ADDUPDT	BP更新フィード
	(ADDWCLR)	(待ち取り消し)
削 除	DEL	下位継続
	DELFIN	削除完了
	DELFAIL	削除失敗
	DELUPDT	BP更新フィード
	DELRDBP	下位BP参照
	DELWRBP	自BP更新

图 3 2

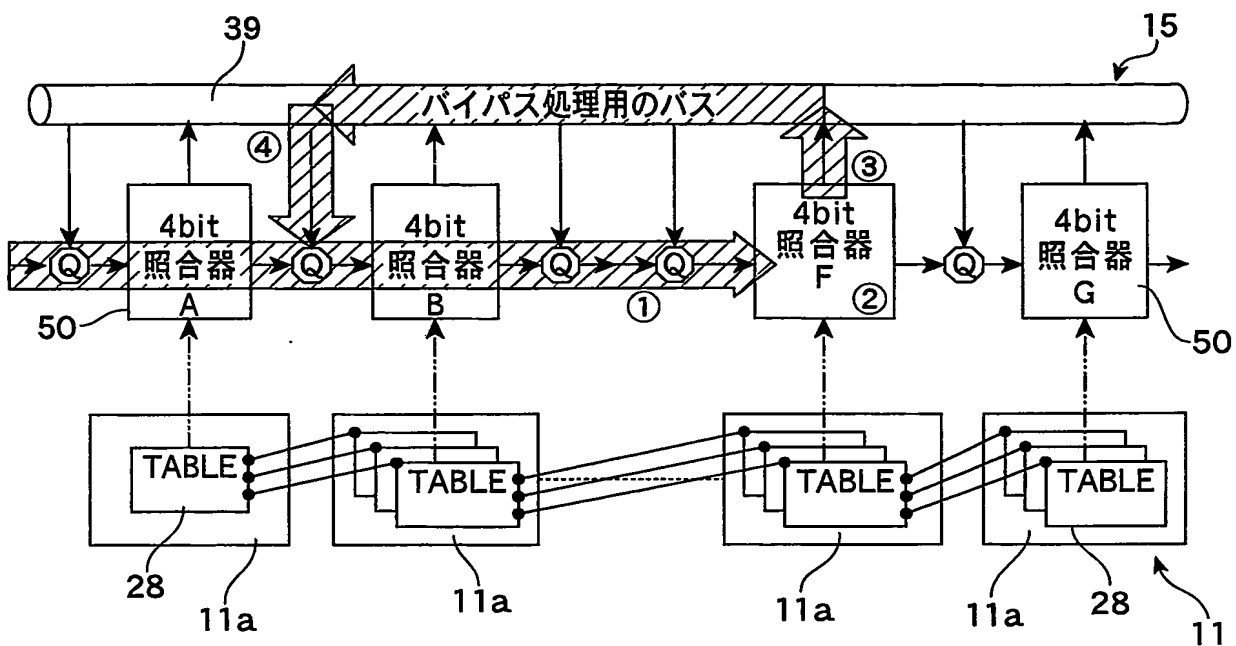


图 3 3

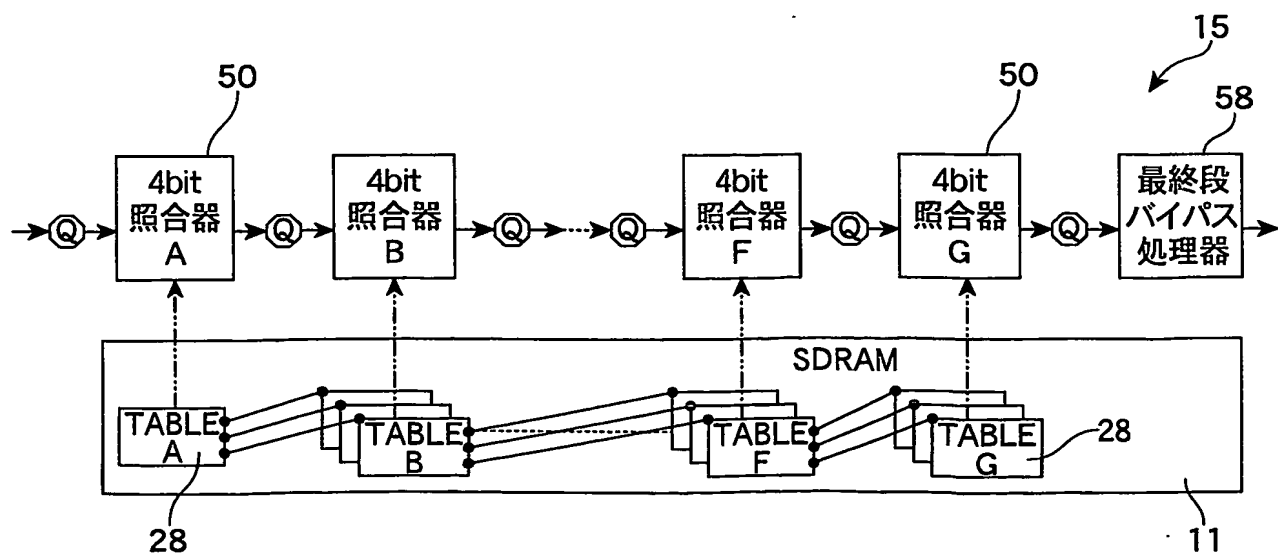


图 3 4

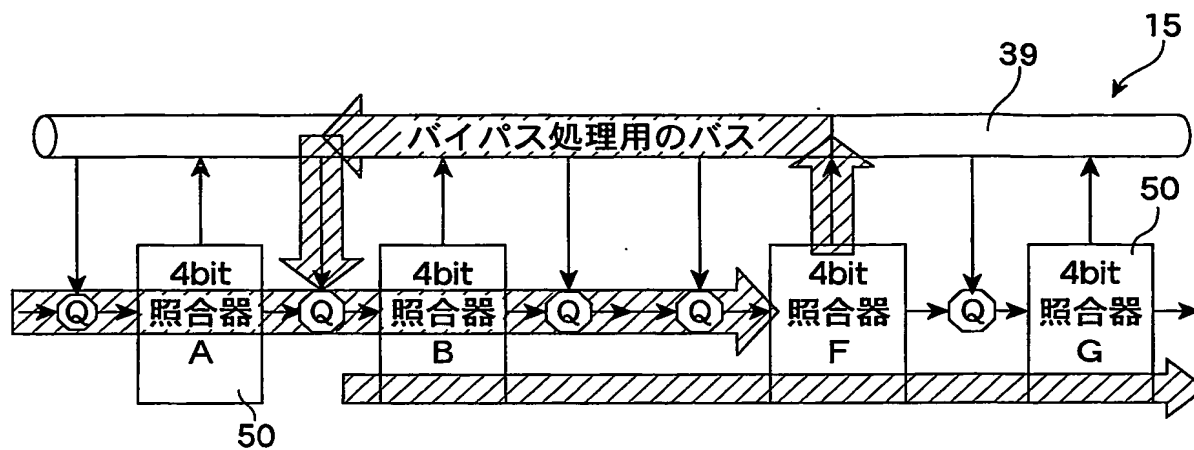
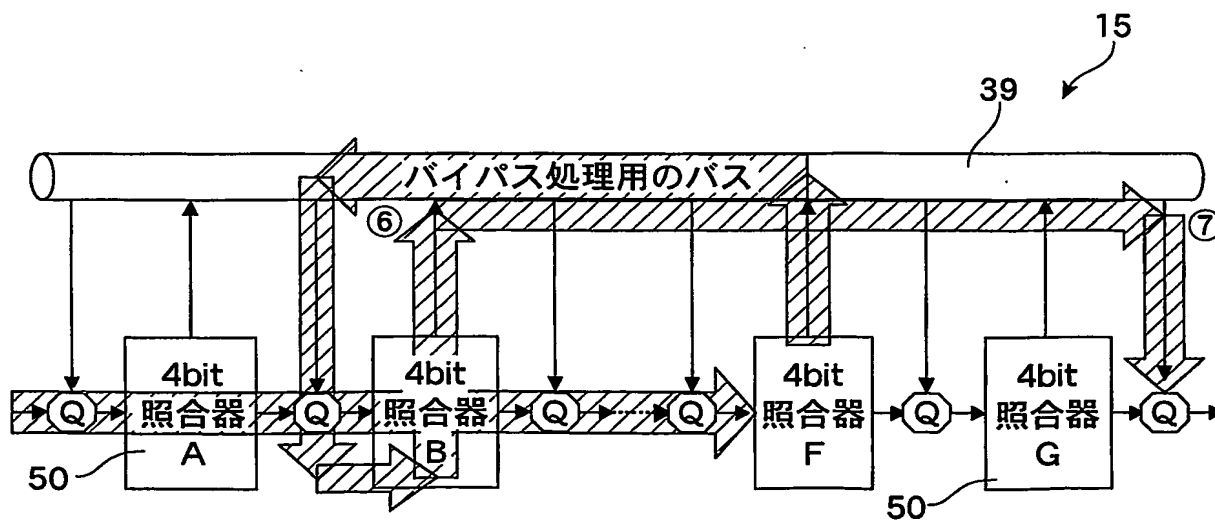


图 3 5



24 / 53

図 36

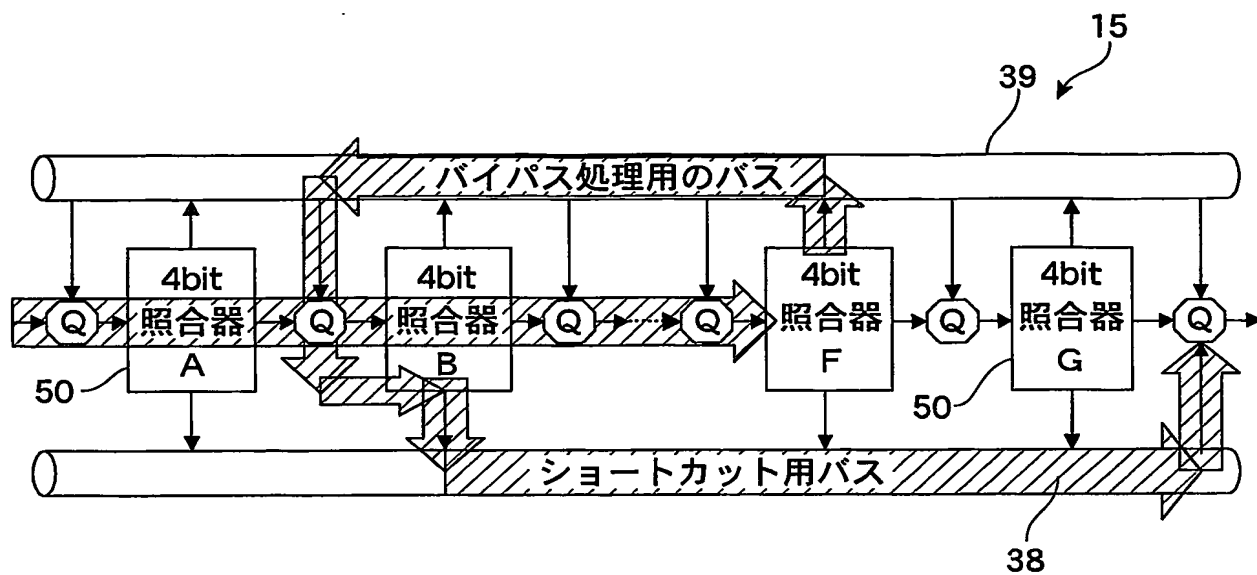


図 37

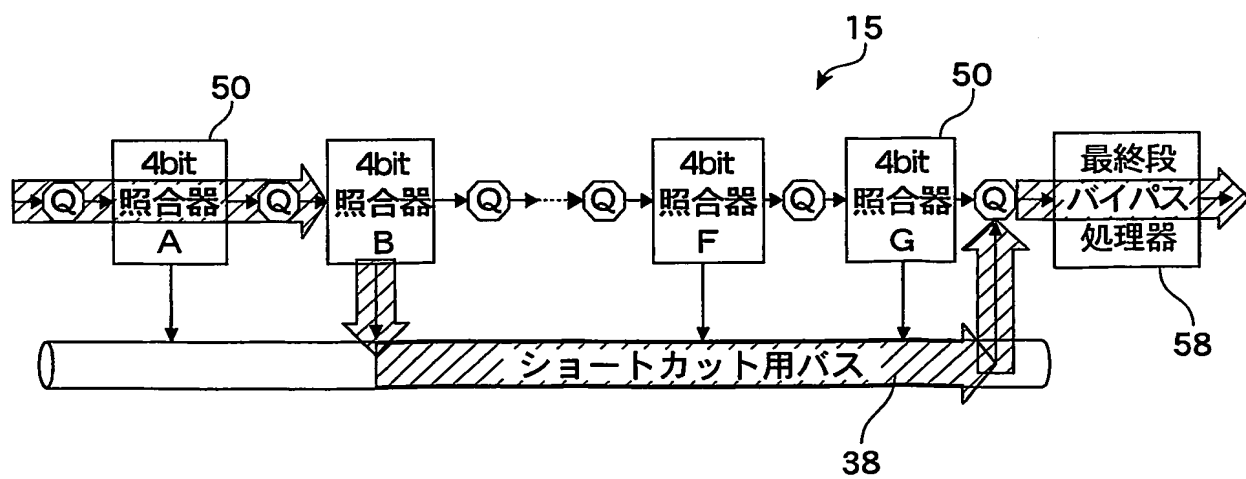


图 38

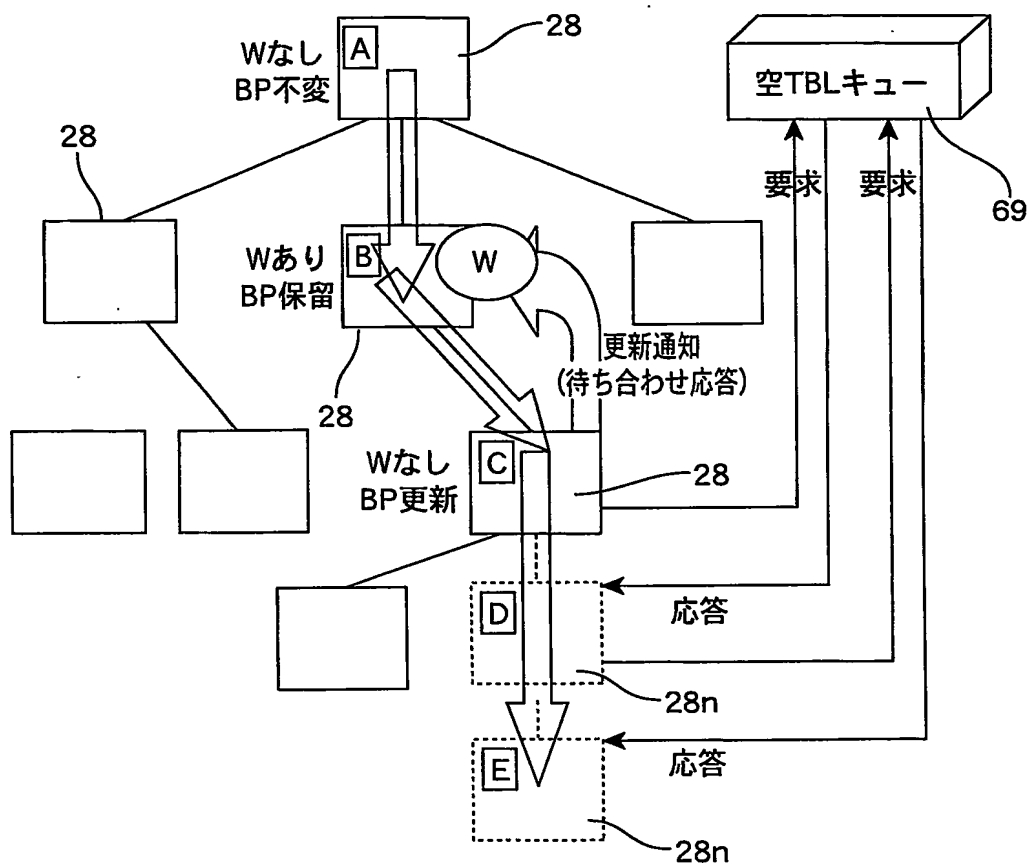
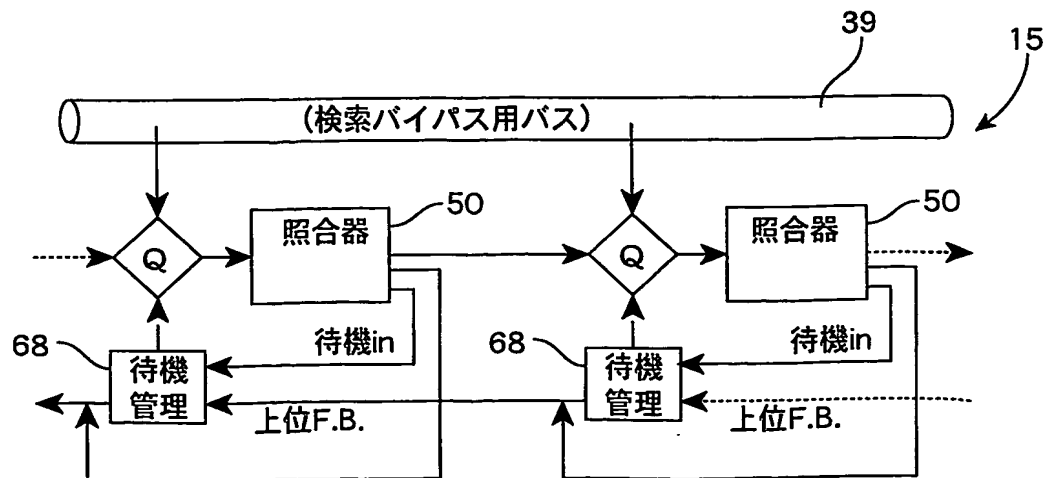
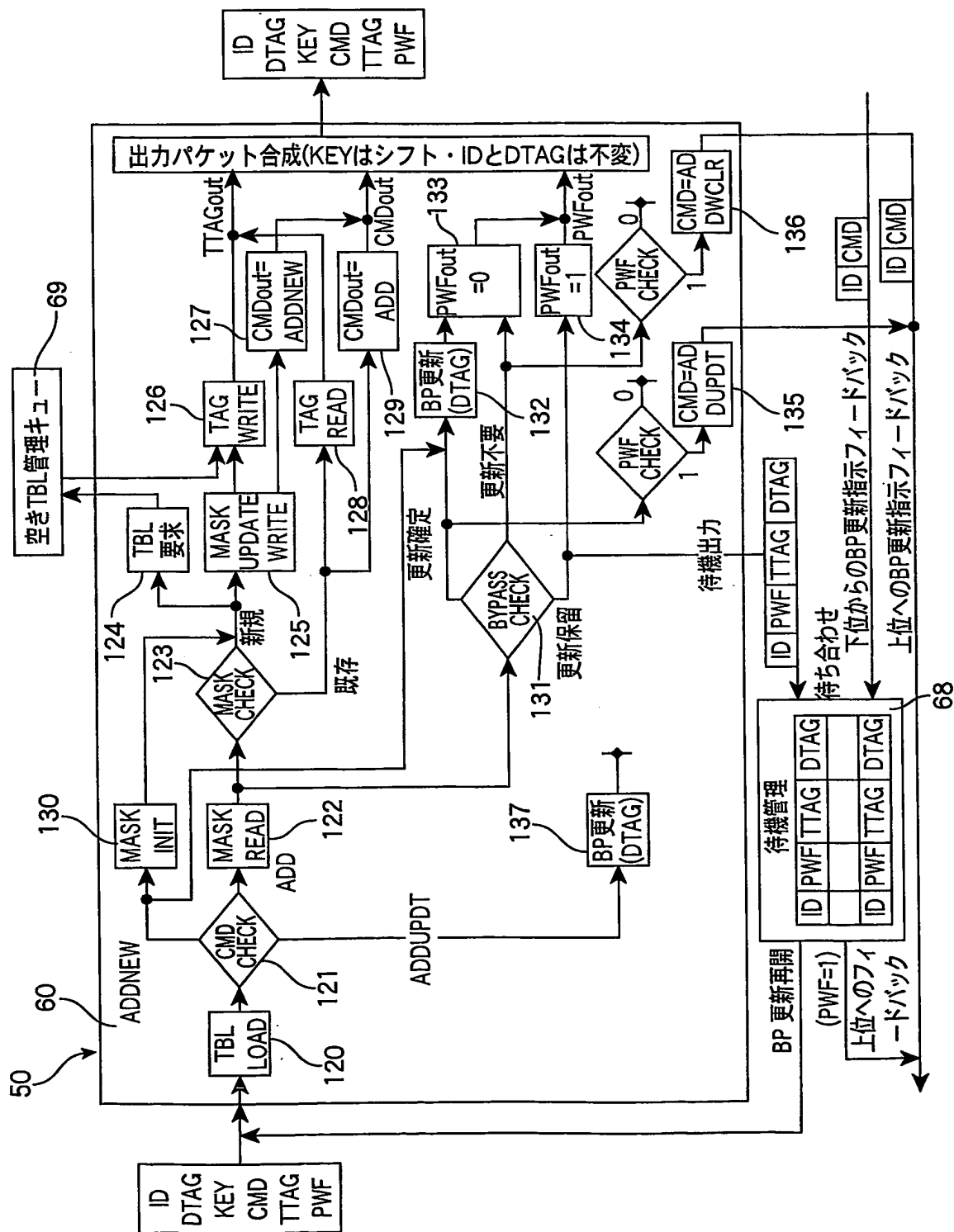


图 39

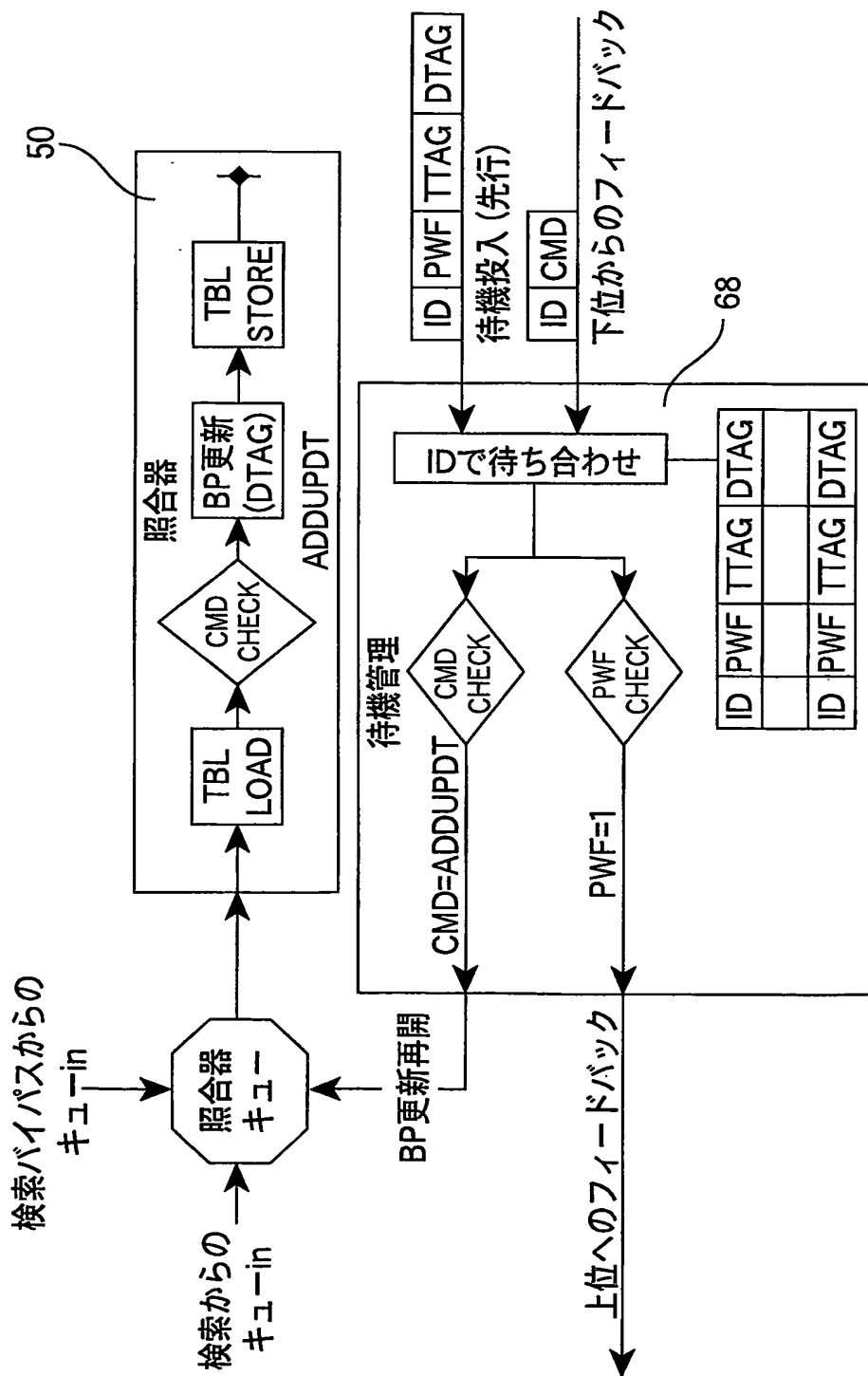


40



27 / 53

図 4 1



28 / 53

図 4 2

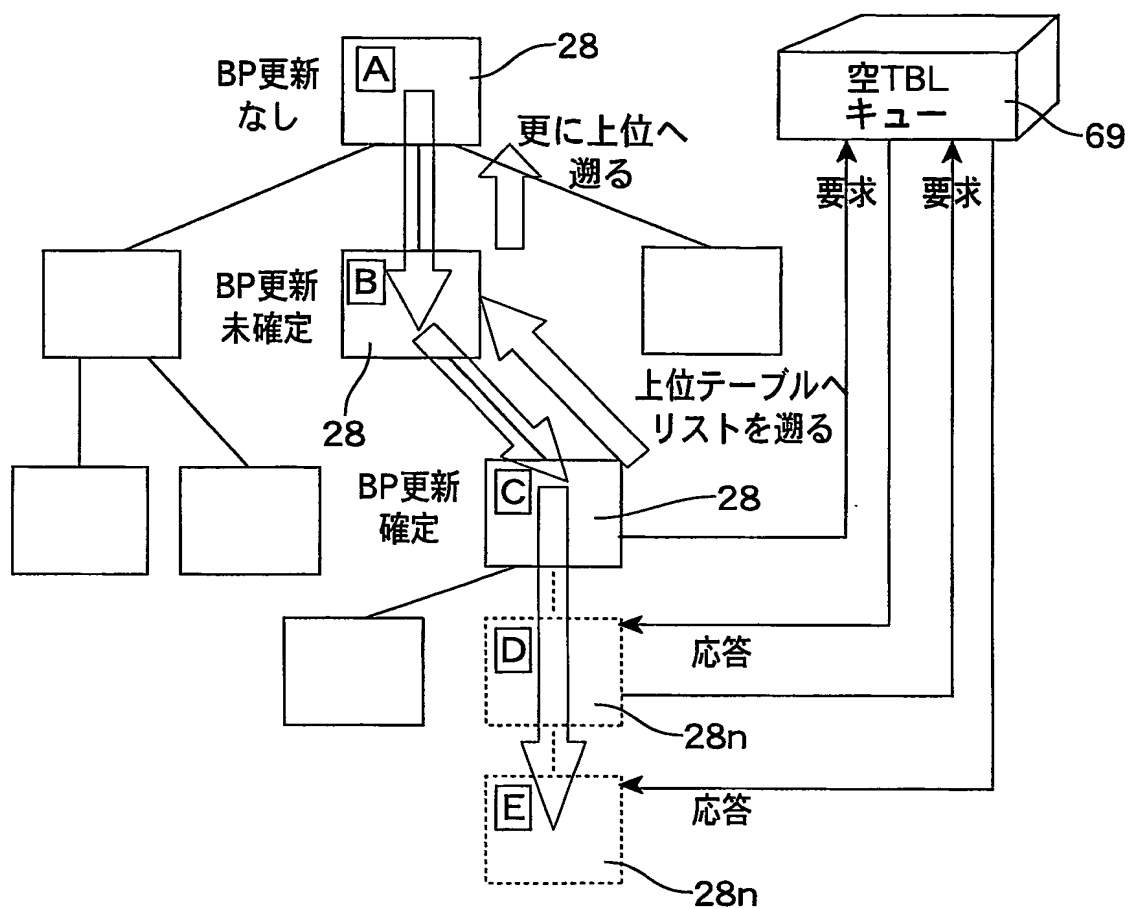
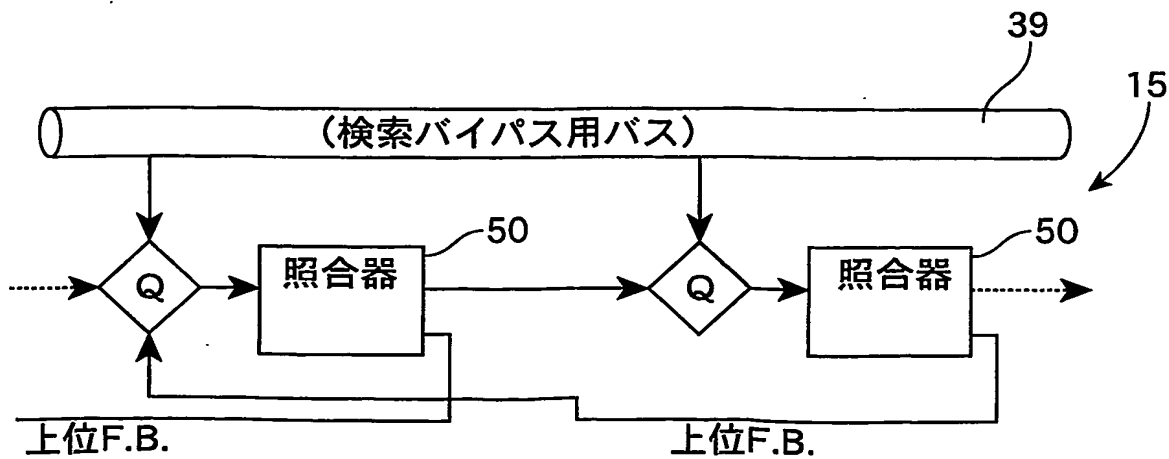


図 4 3



29 / 53

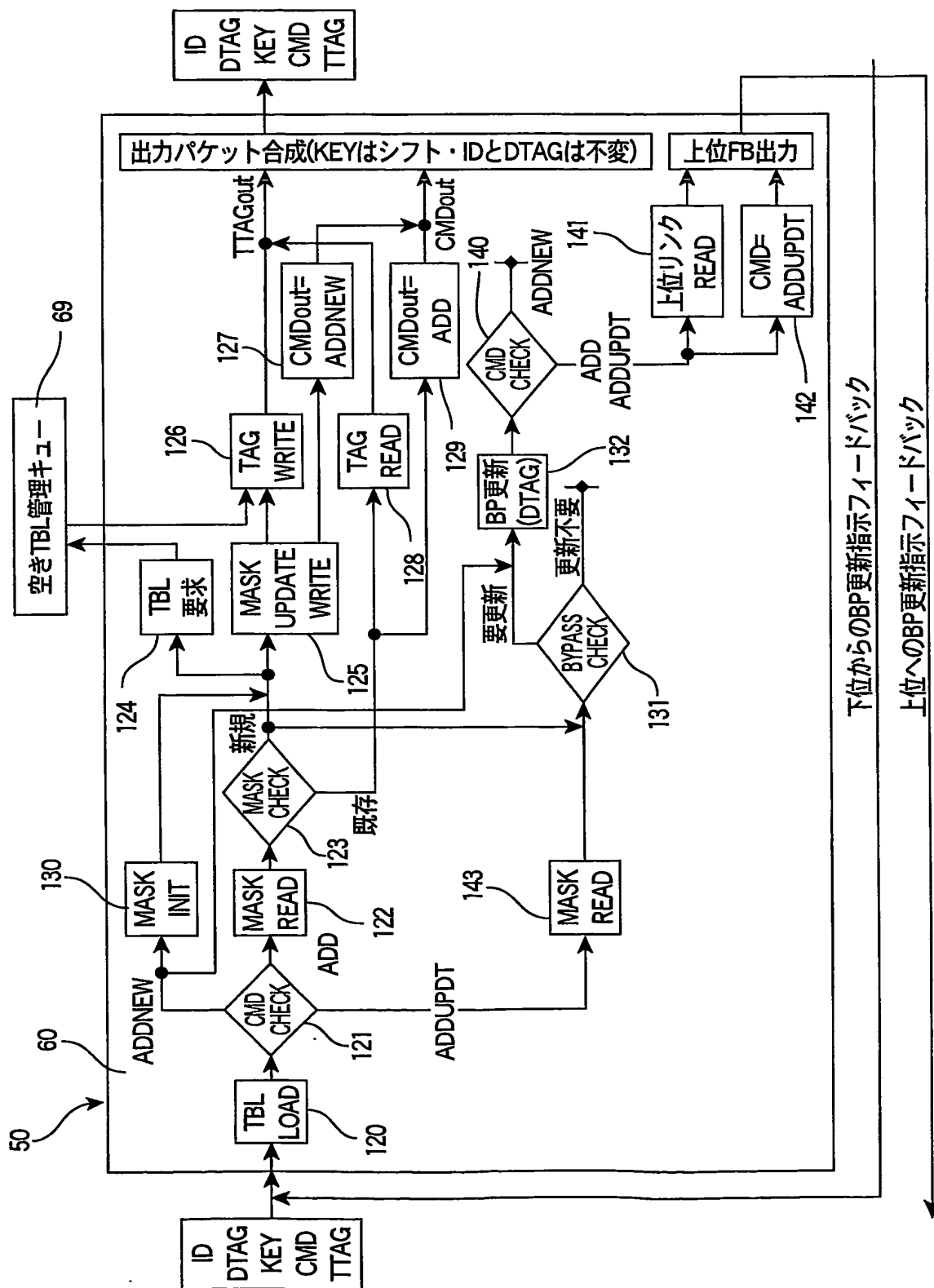
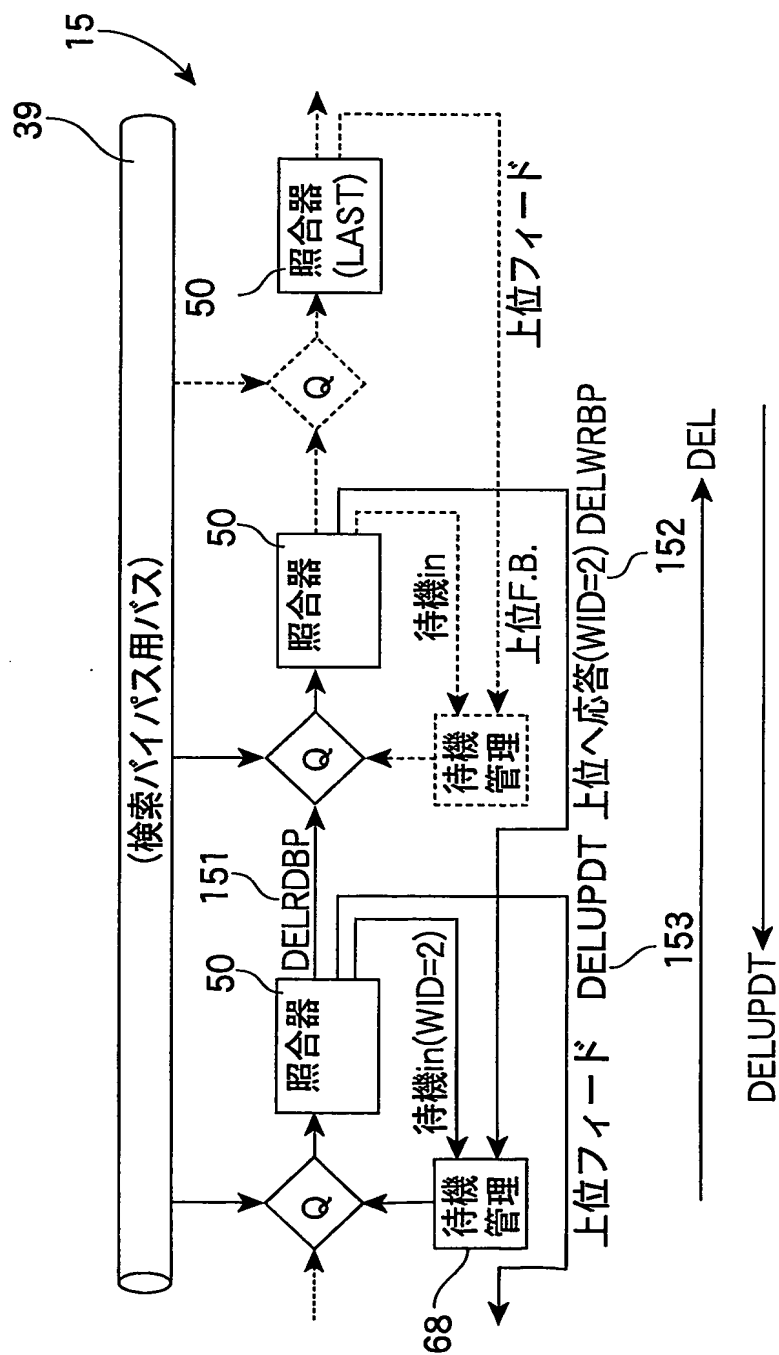


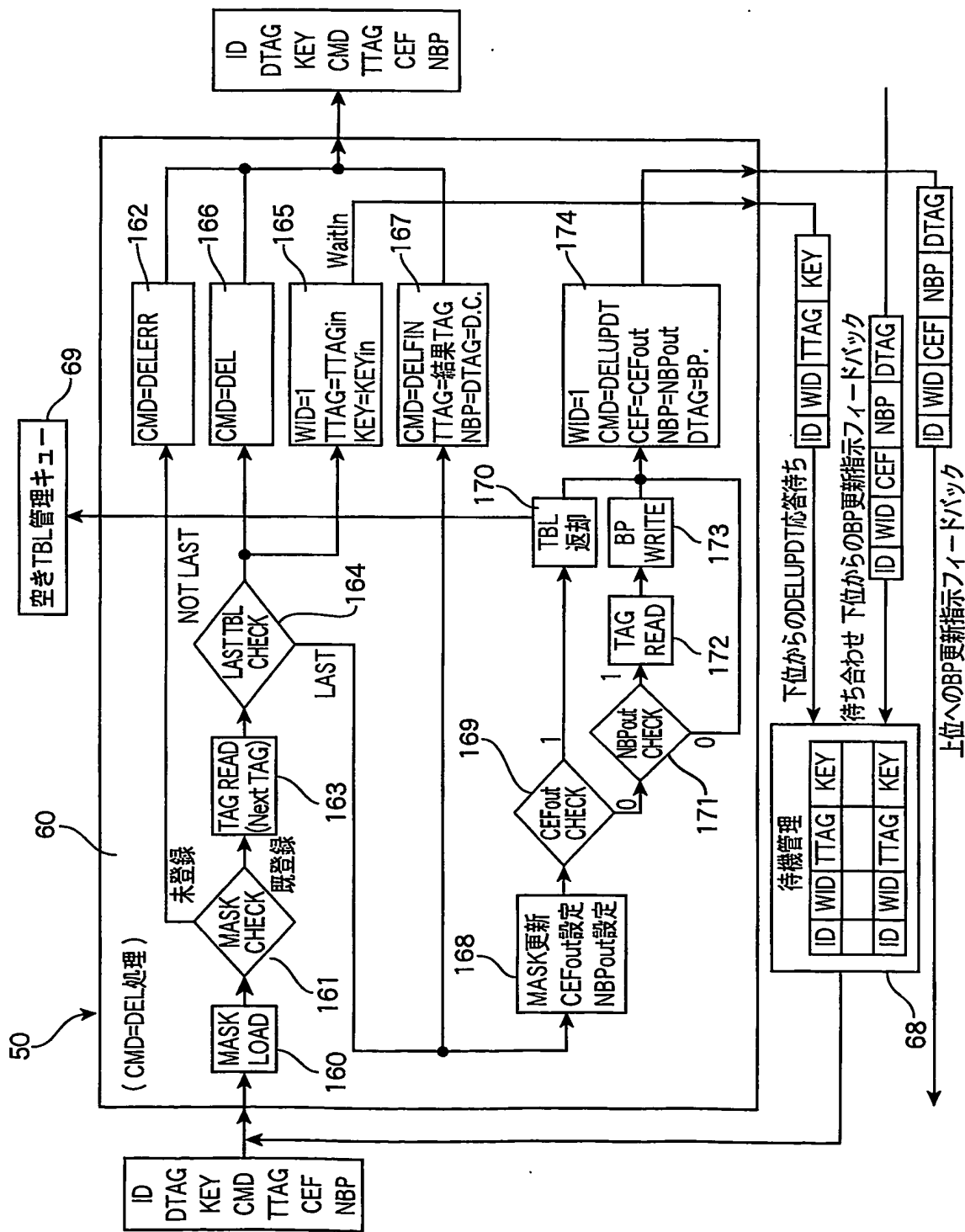
図 44

54X

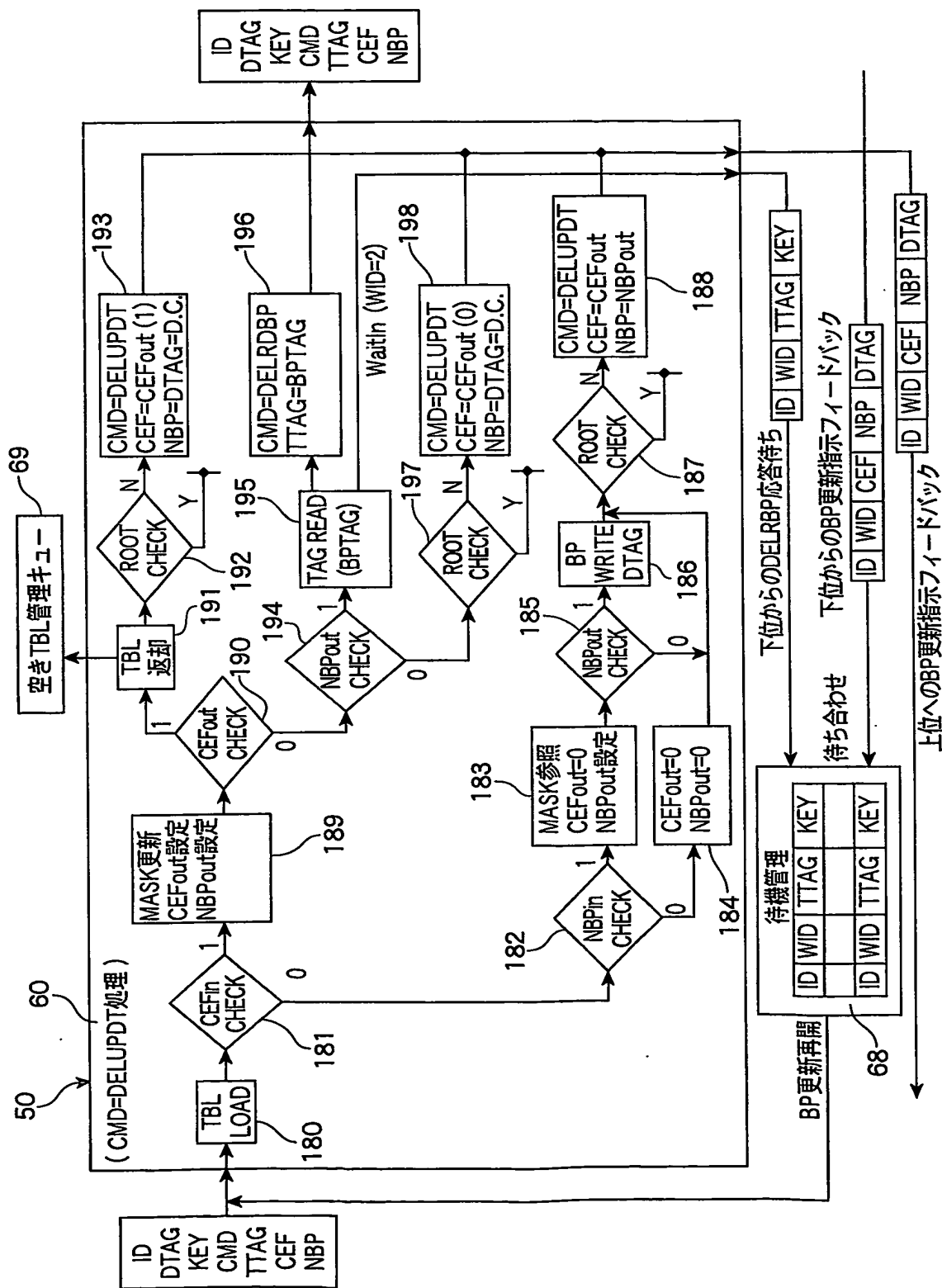


3 1 / 5 3

図 4 6

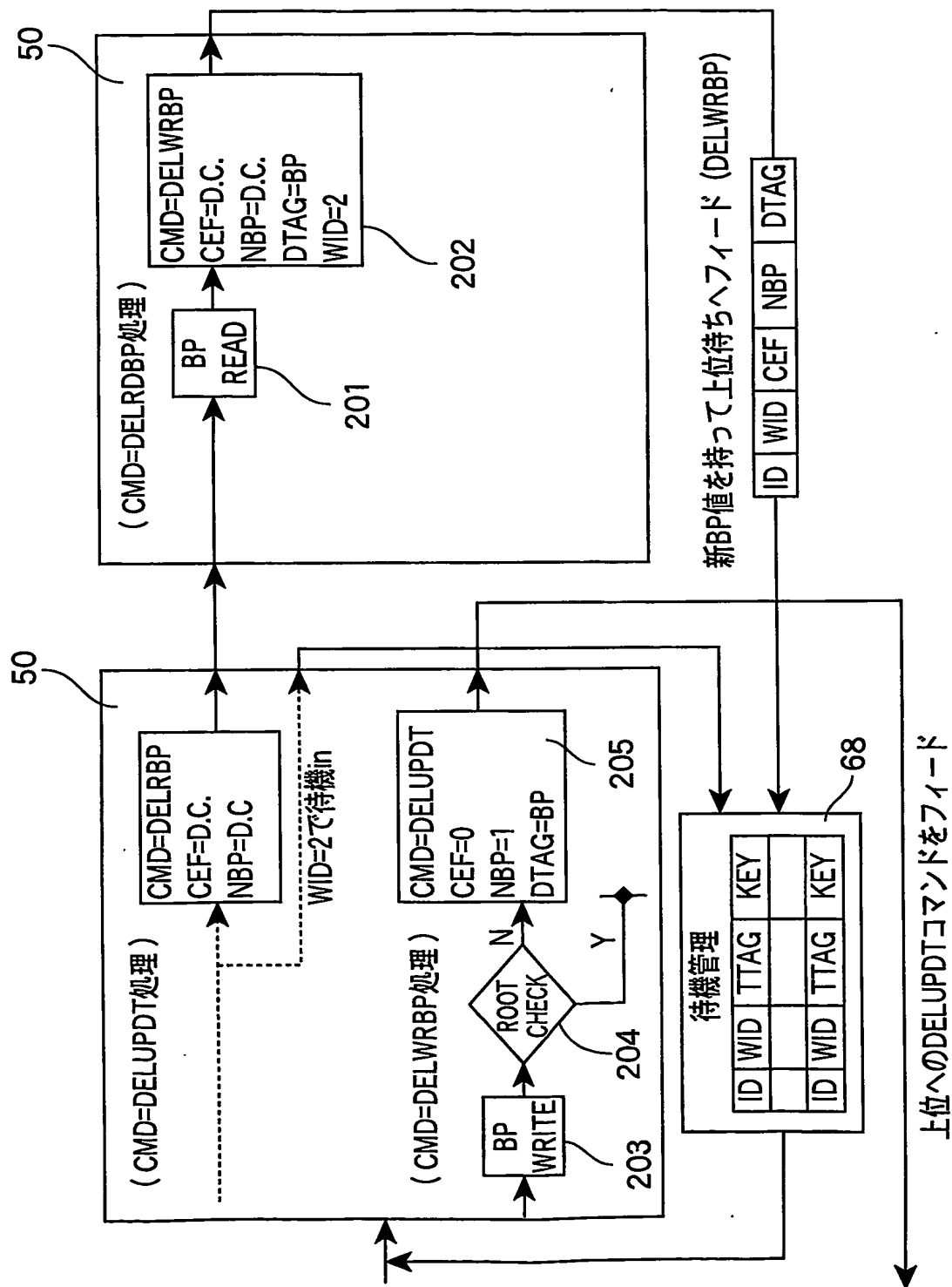


74图



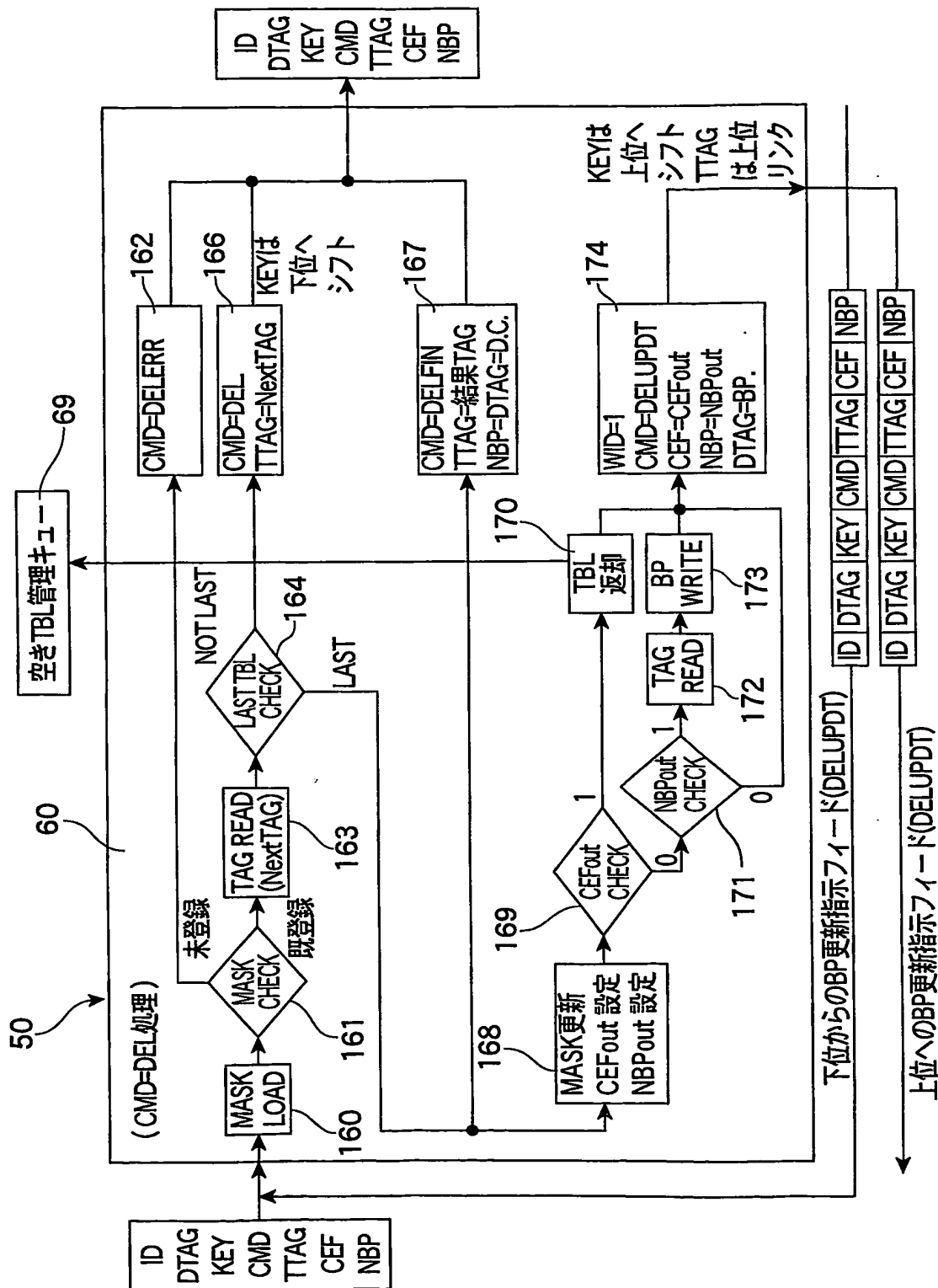
3 3 / 5 3

図 4 8



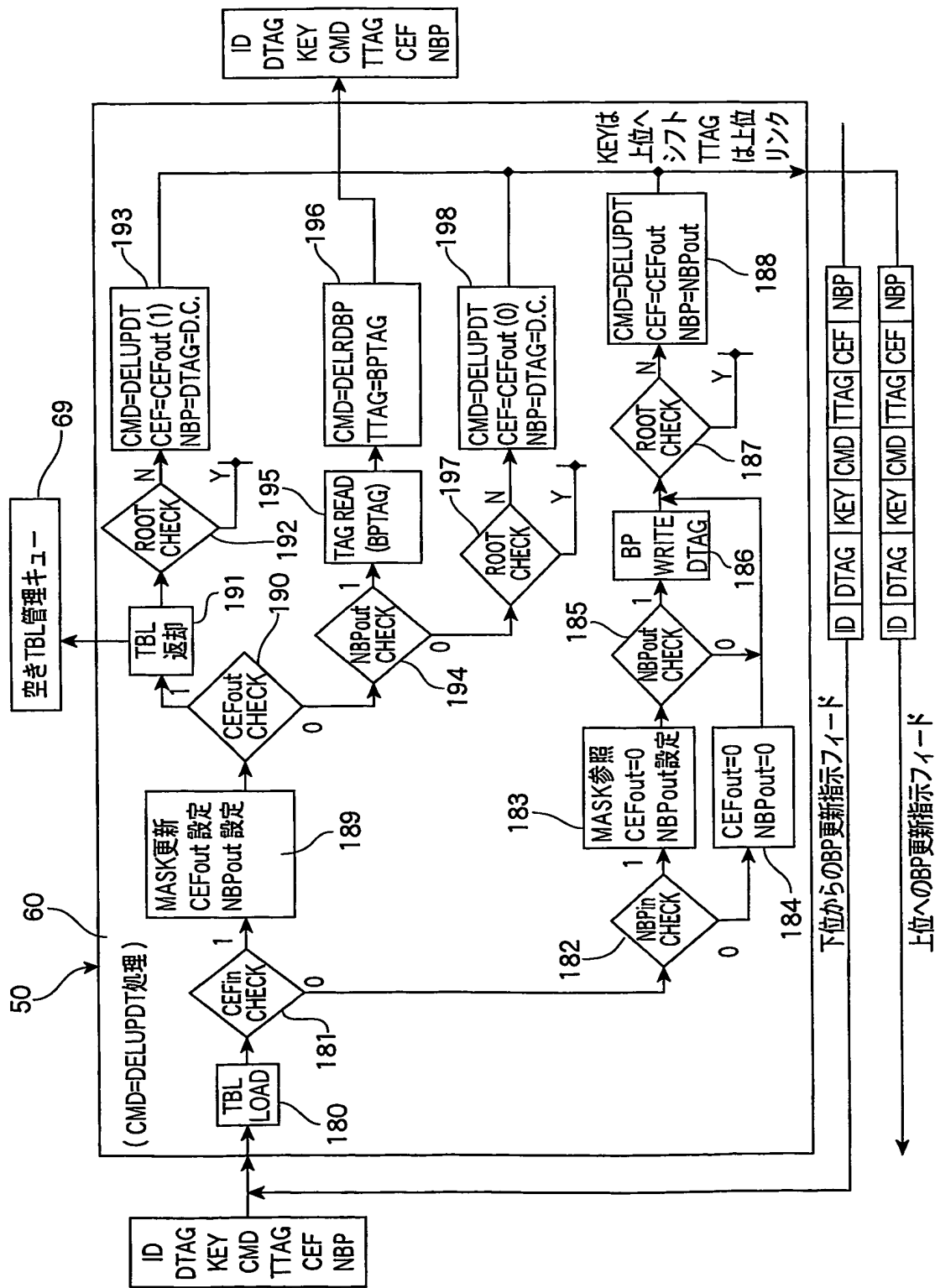
3 4 / 5 3

図 4 9



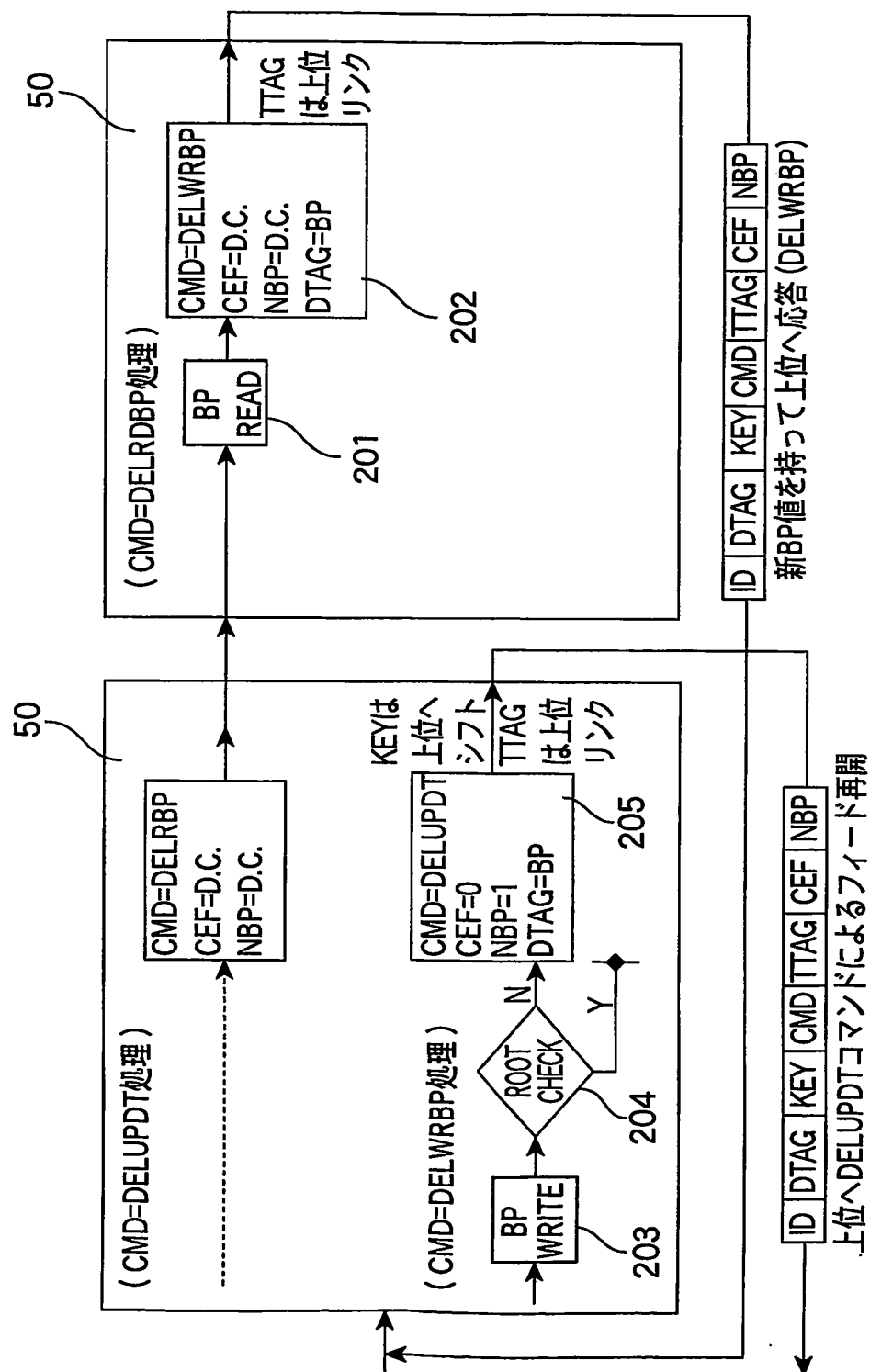
3 5 / 5 3

図 50



36 / 53

図 5 1



37 / 53

図 5 2

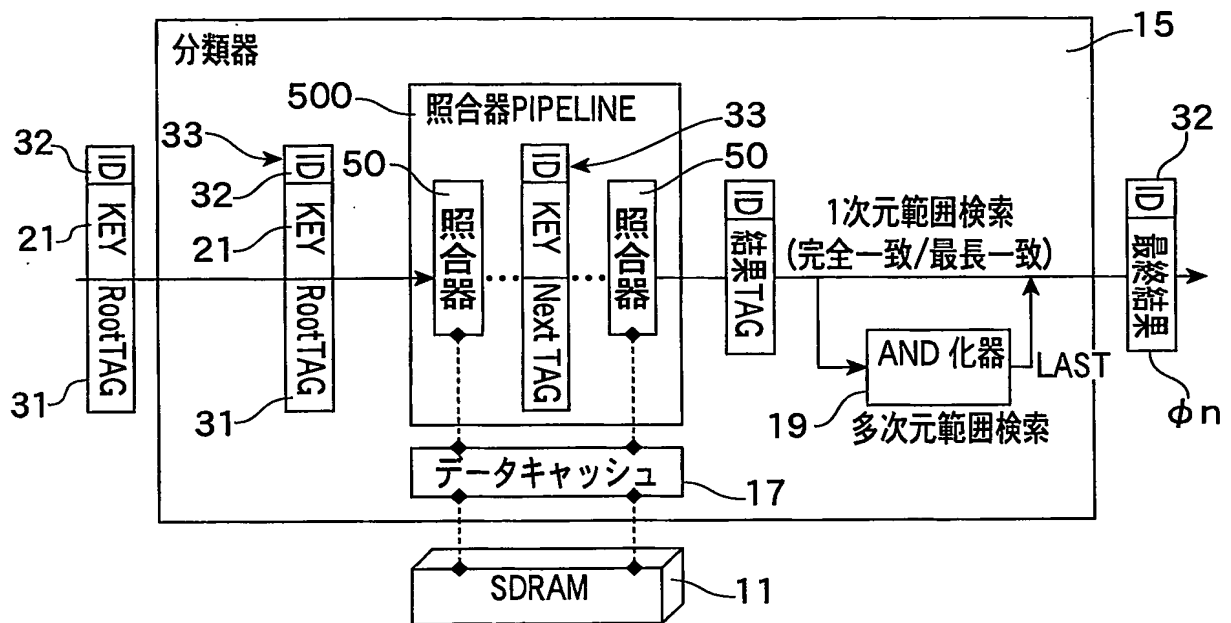
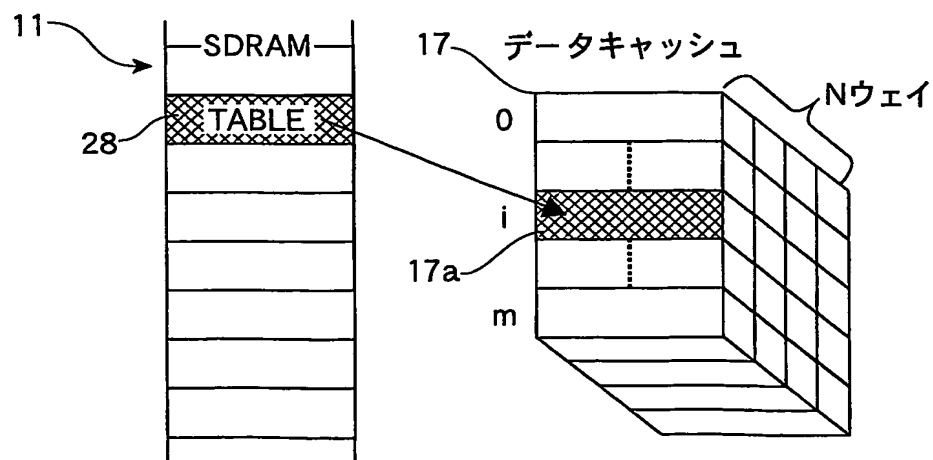
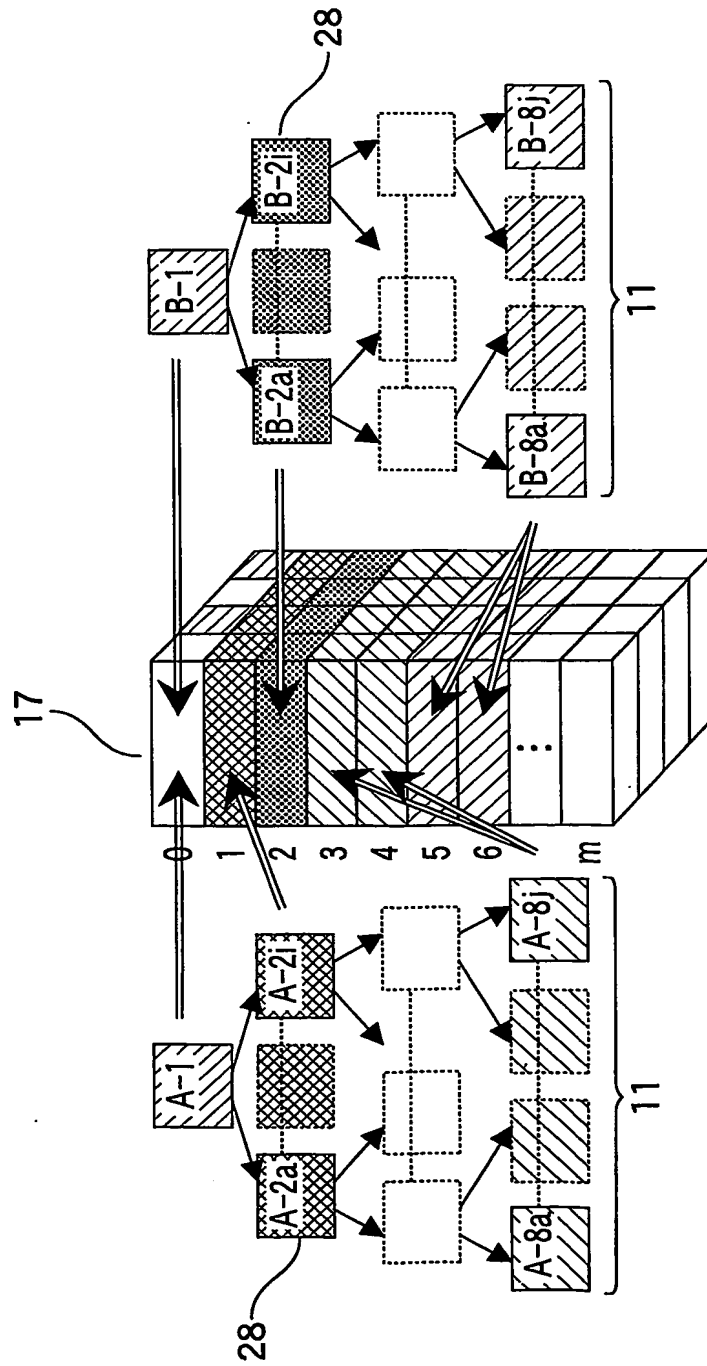


図 5 3



3 8 / 5 3

図 5 4



39 / 53

図 5 5

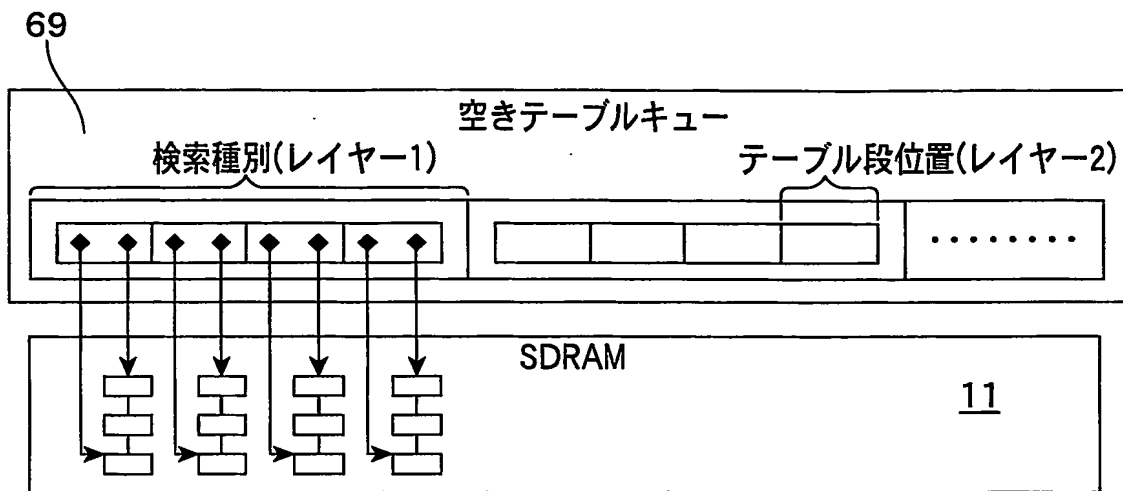
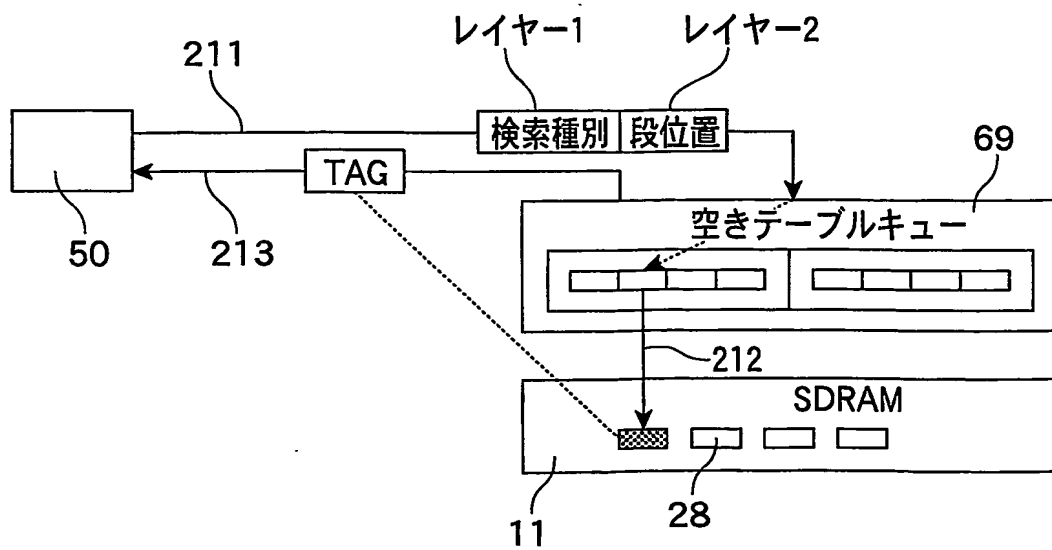
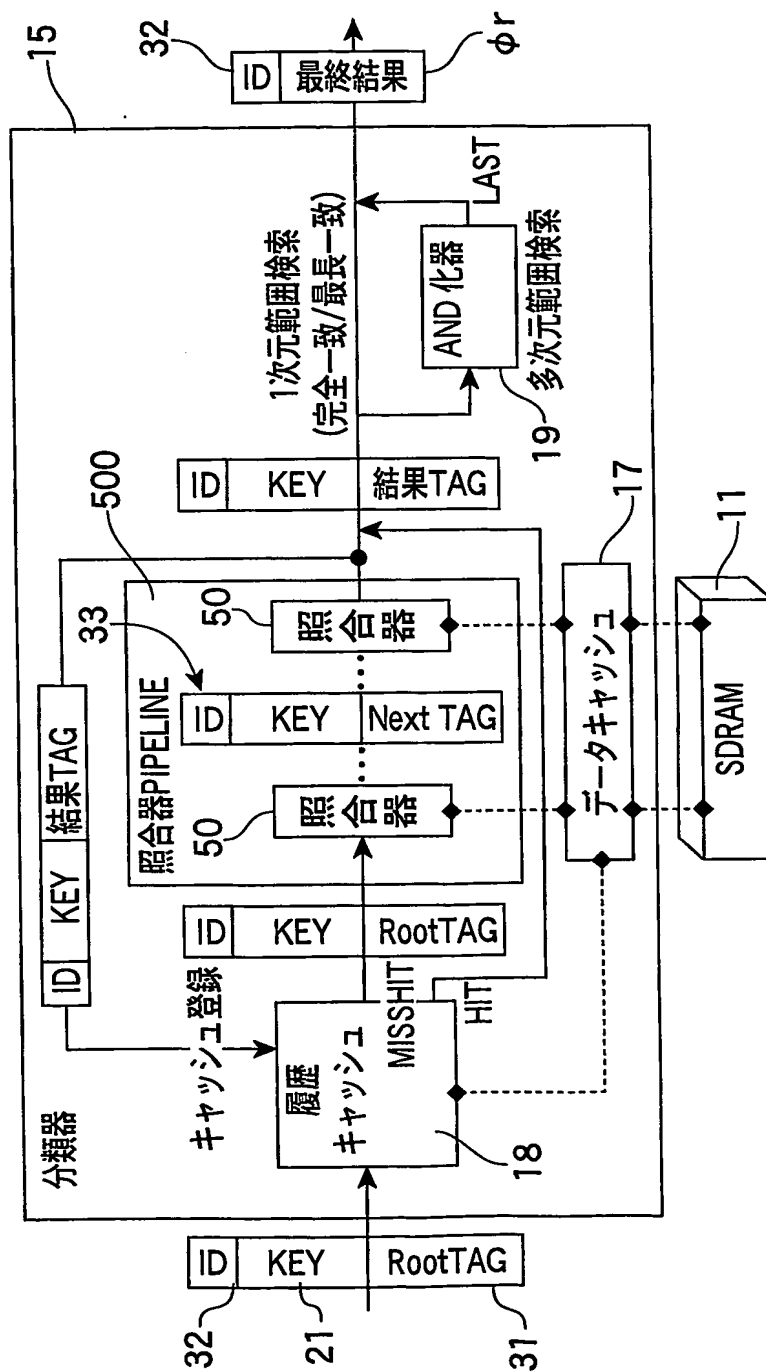


図 5 6



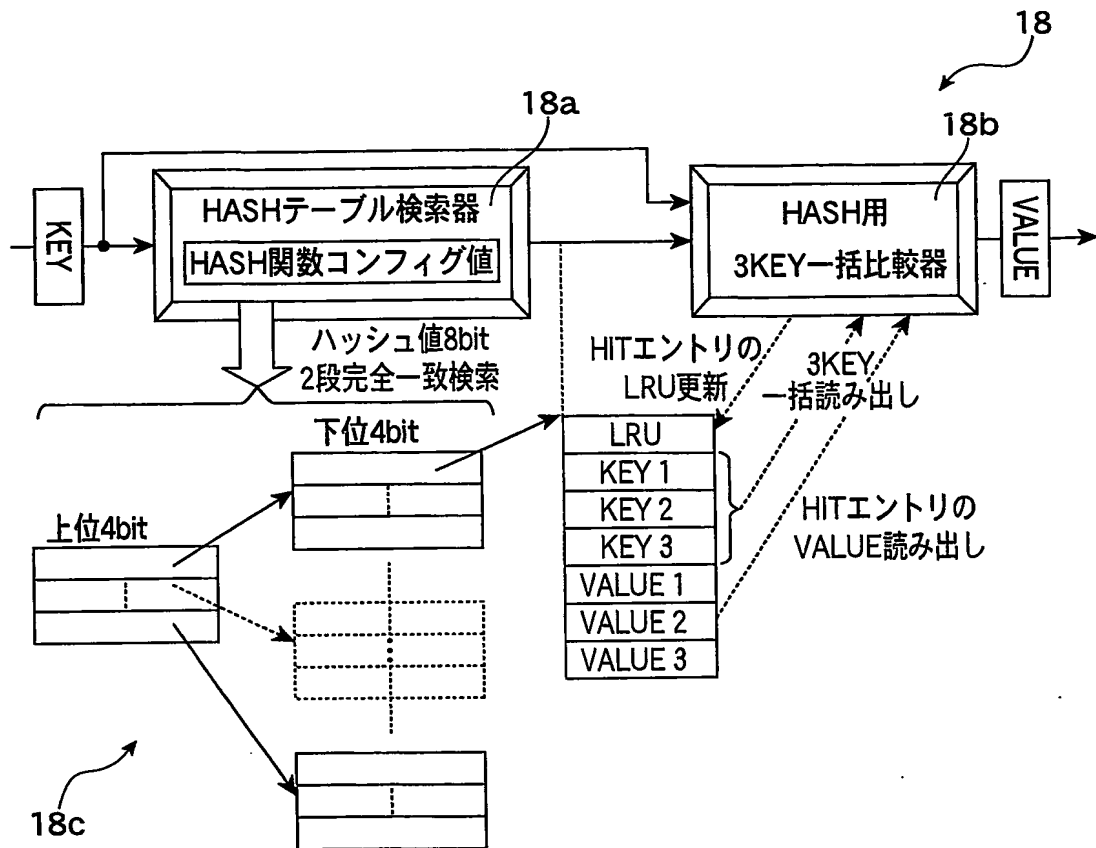
40 / 53



75X

4 1 / 5 3

図 5 8



4 2 / 5 3

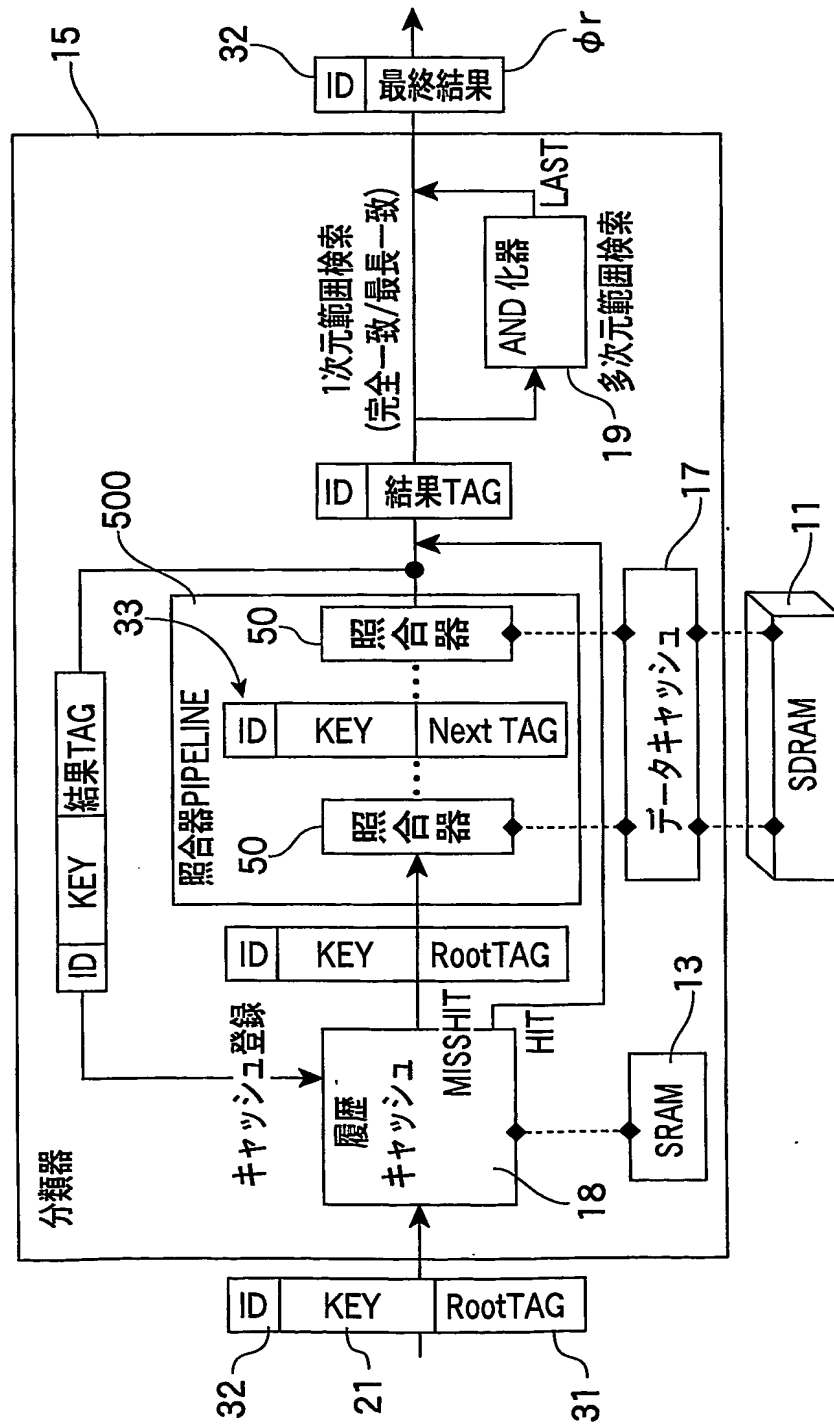
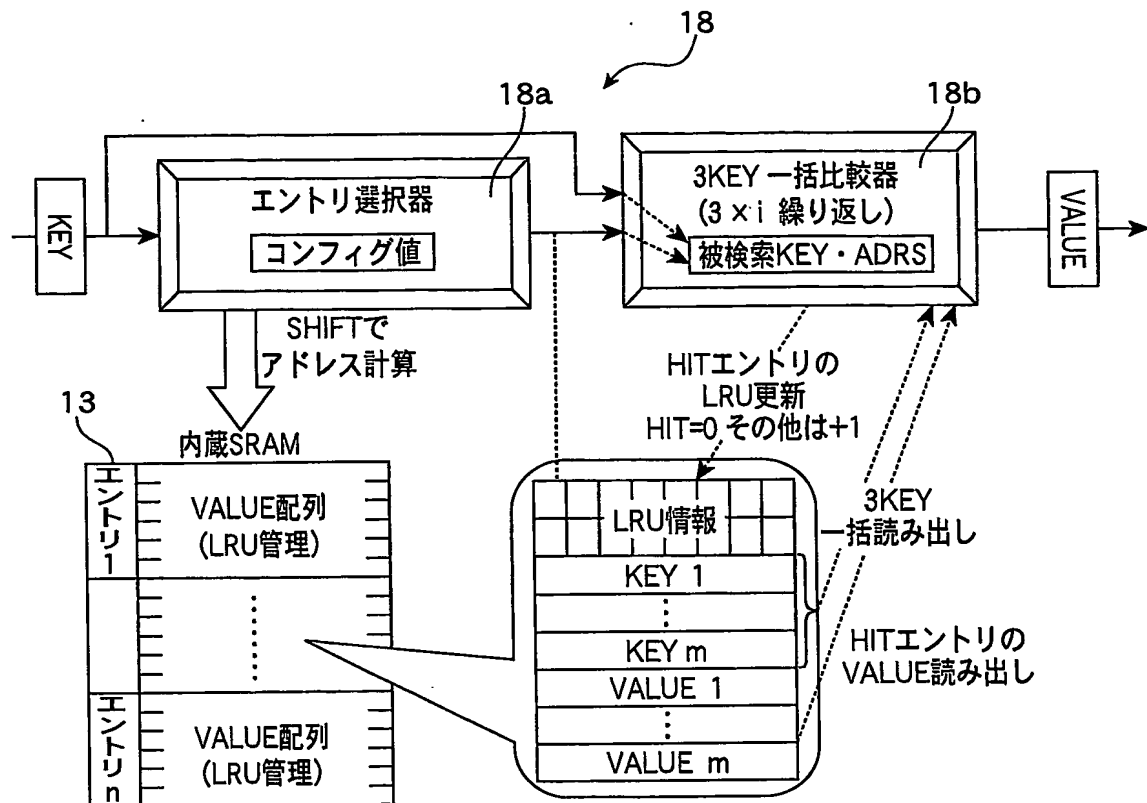


図 5 9

43 / 53

図 60



4 4 / 5 3

図 6 1

$$\begin{array}{c}
 \text{行列 A} \\
 \left[\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{array} \right] \times \begin{array}{c} \text{行列 B} \\ \left[\begin{array}{ccc} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{array} \right]
 \end{array}
 \end{array}$$

$$= \begin{array}{c}
 \begin{array}{c} \text{A1} \times \text{B1} \\ \left[\begin{array}{c} a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} \\ a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} \end{array} \right] \\ \text{A2} \times \text{B1} \end{array} \\
 \text{行列 Z} \\
 \begin{array}{c} \text{A1} \times \text{B2} \\ \left[\begin{array}{c} a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32} \\ a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32} \end{array} \right] \\ \text{A2} \times \text{B2} \\
 \begin{array}{c} \text{A1} \times \text{B3} \\ \left[\begin{array}{c} a_{11} * b_{13} + a_{12} * b_{23} + a_{13} * b_{33} \\ a_{21} * b_{13} + a_{22} * b_{23} + a_{23} * b_{33} \end{array} \right] \\ \text{A2} \times \text{B3} \end{array}
 \end{array}
 \end{array}$$

図 6 2

$$\begin{array}{c}
 \text{行列 Z} \\
 \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right]
 \end{array}
 \begin{array}{l}
 \longrightarrow 1 \text{ or } 0 \text{ or } 0 = 1 \\
 \longrightarrow 0 \text{ or } 0 \text{ or } 0 = 0
 \end{array}$$

図 6 3

$$\begin{array}{c}
 \text{行列 X} \\
 \left[\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{array} \right] \times \begin{array}{c} \text{行列 Y} \\ \left[\begin{array}{ccc} b_{11} & b_{12} & \dots & c_{1i} & \dots \\ b_{21} & b_{22} & \dots & c_{2i} & \dots \\ b_{31} & b_{32} & \dots & c_{3i} & \dots \end{array} \right]
 \end{array}
 \end{array}$$

$$= \begin{array}{c}
 \begin{array}{c} \text{AとB1の演算結果} \\ \left[\begin{array}{c} a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} \\ a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} \end{array} \right] \\ \text{AとB2の演算結果} \\ \left[\begin{array}{c} a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32} \\ a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32} \end{array} \right] \\
 \dots \\
 \left[\begin{array}{c} a_{11} * c_{1i} + a_{12} * c_{2i} + a_{13} * c_{3i} \\ a_{21} * c_{1i} + a_{22} * c_{2i} + a_{23} * c_{3i} \end{array} \right] \\
 \dots \\
 \text{AとCiの演算結果}
 \end{array}
 \end{array}$$

4 5 / 5 3

図 6 4

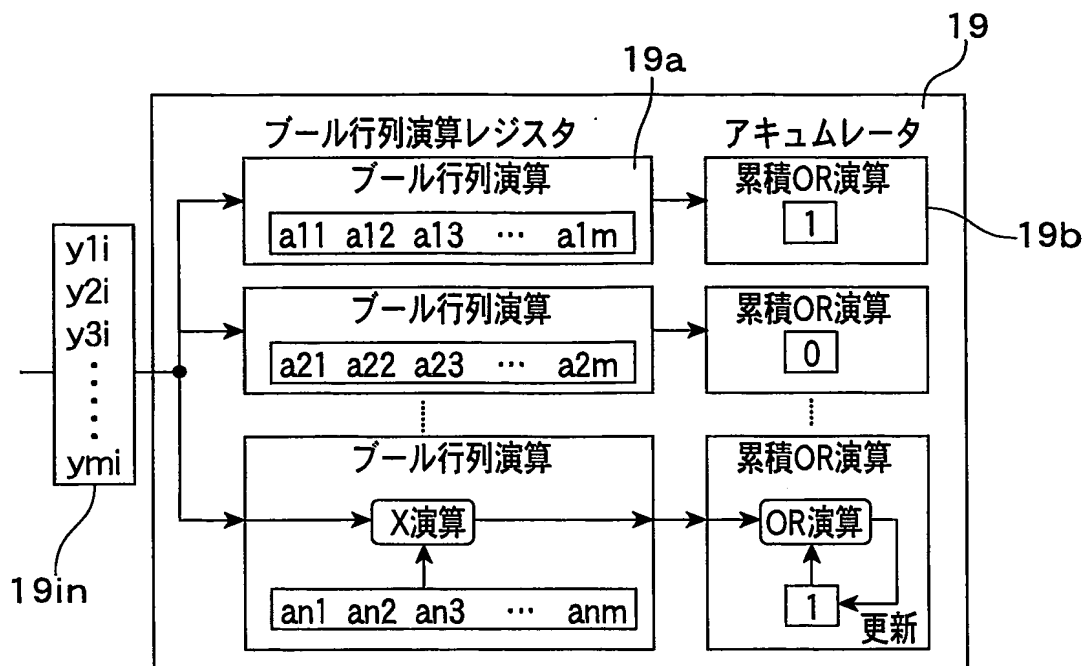
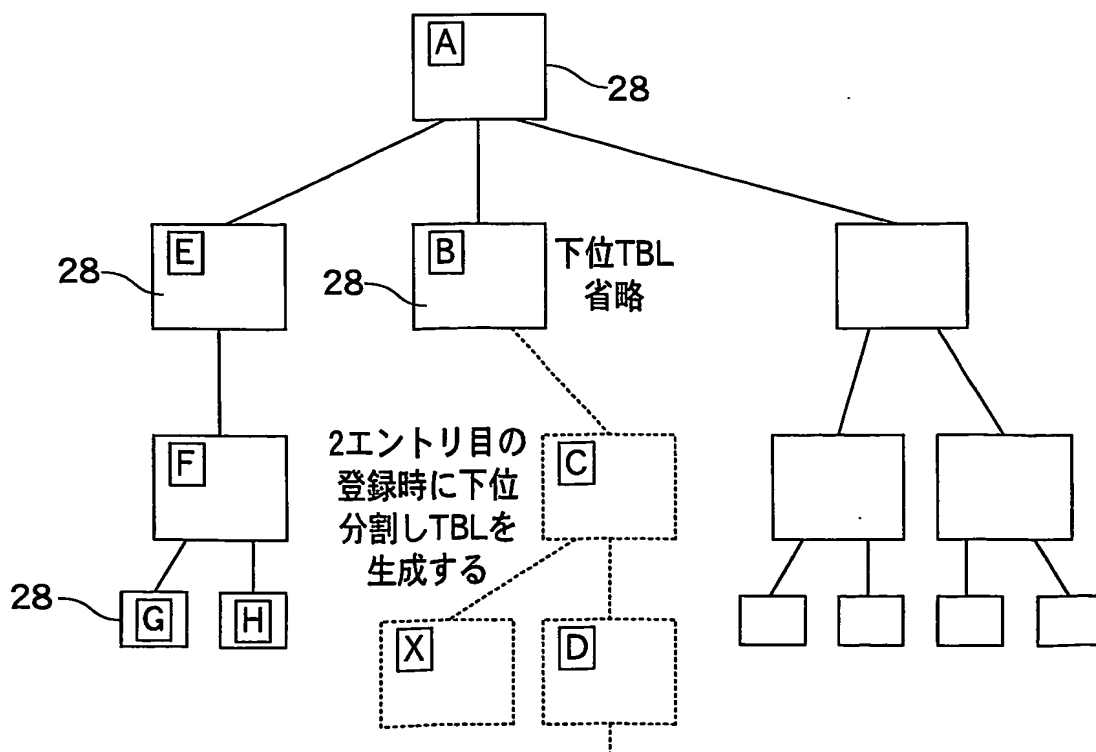


図 6 5



46 / 53

図 6 6

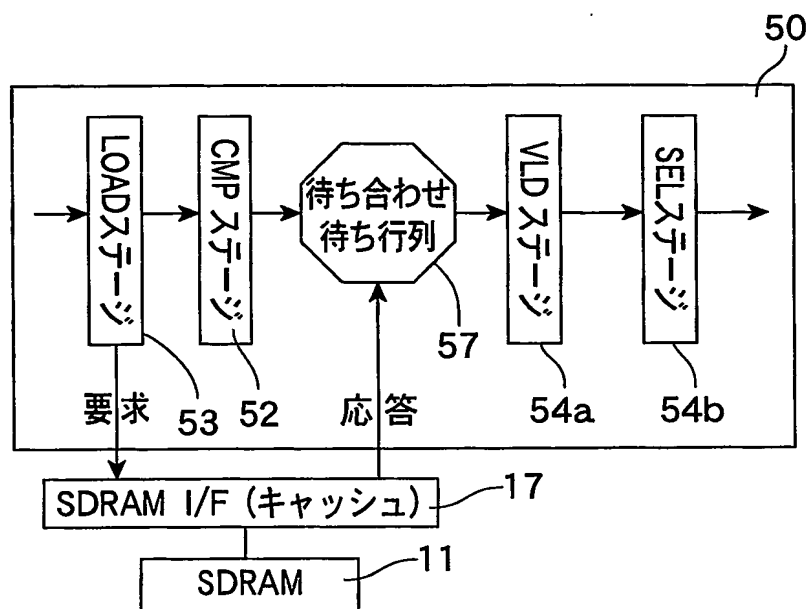
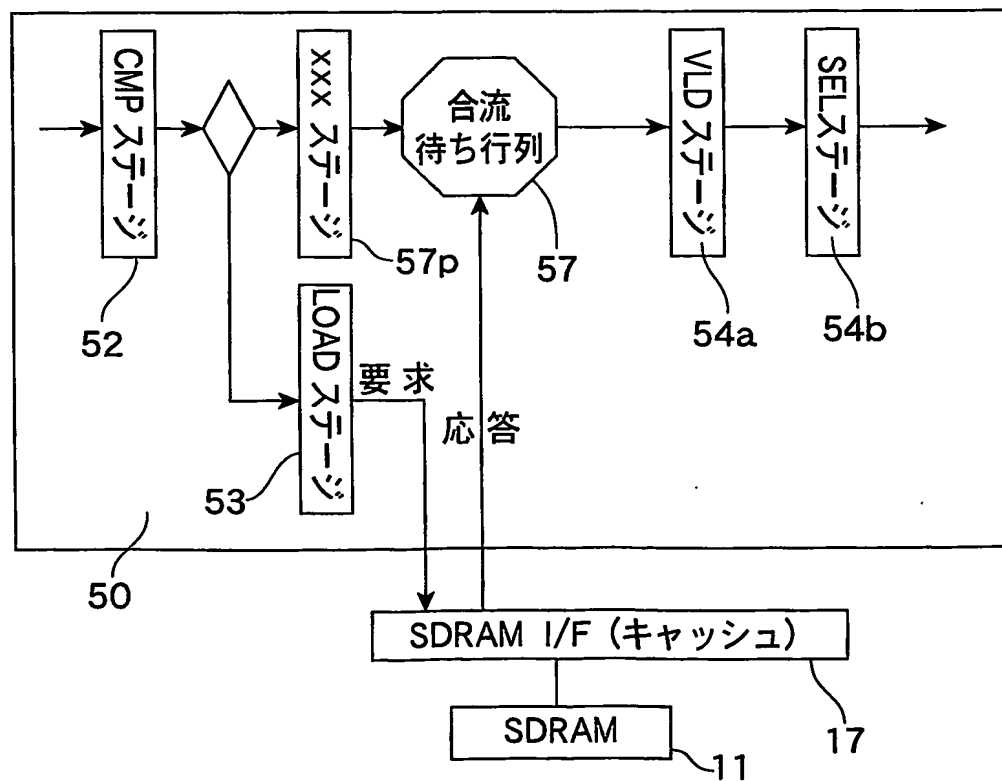


図 6 7



47 / 53

図 6 8

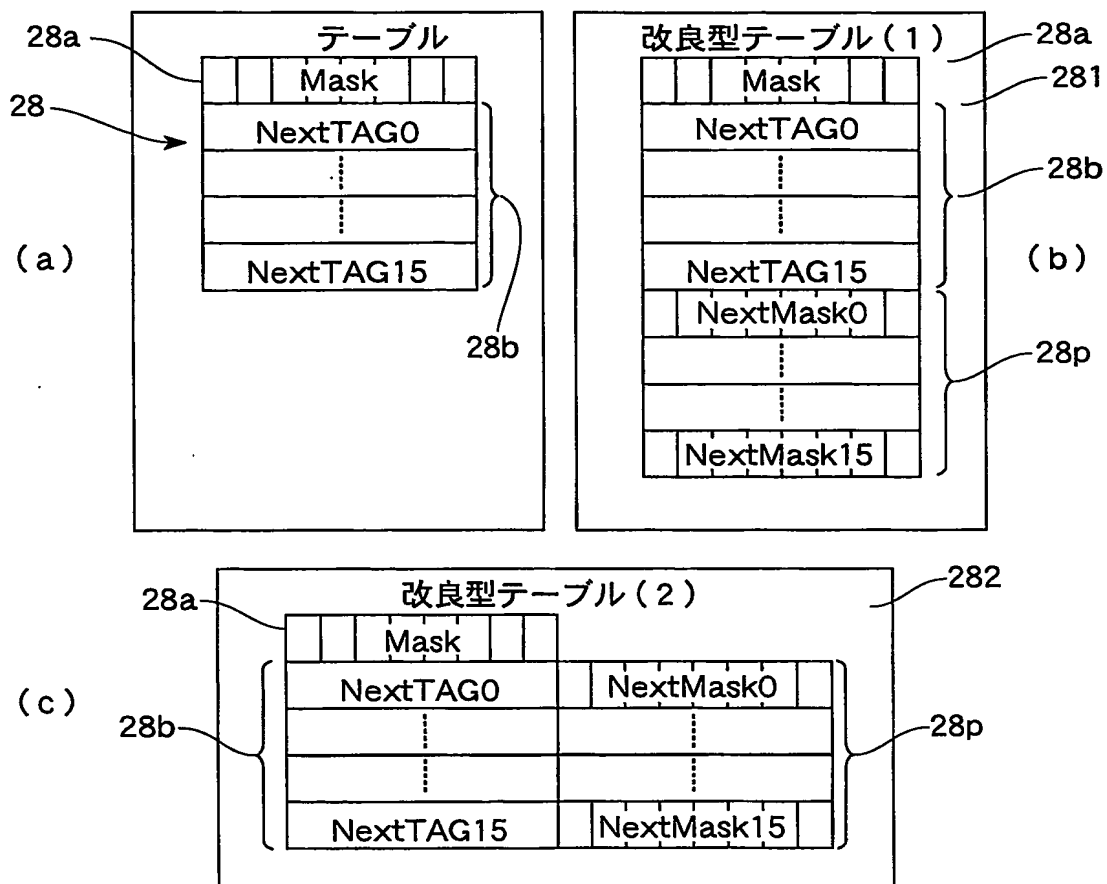
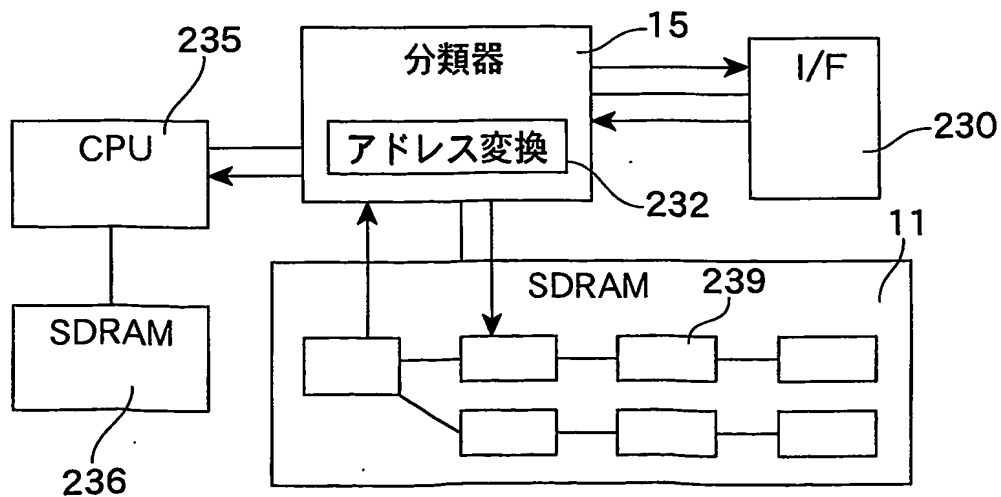


図 6 9



48 / 53

図 7 0

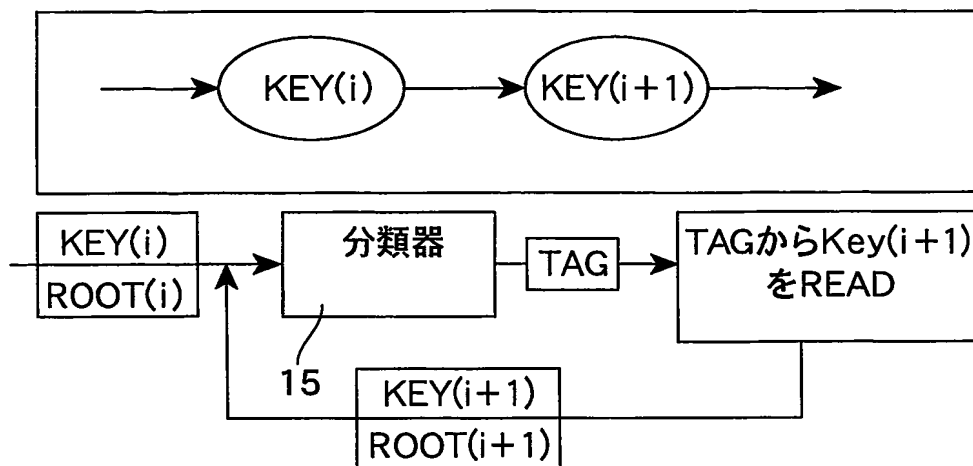
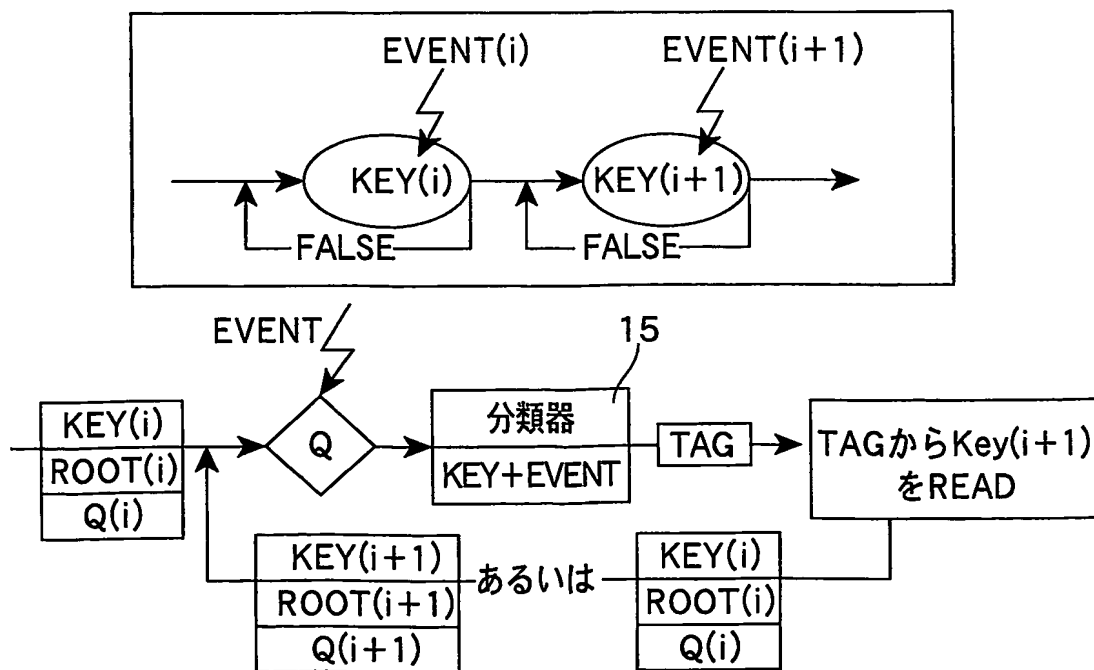


図 7 1



49 / 53

図 7 2

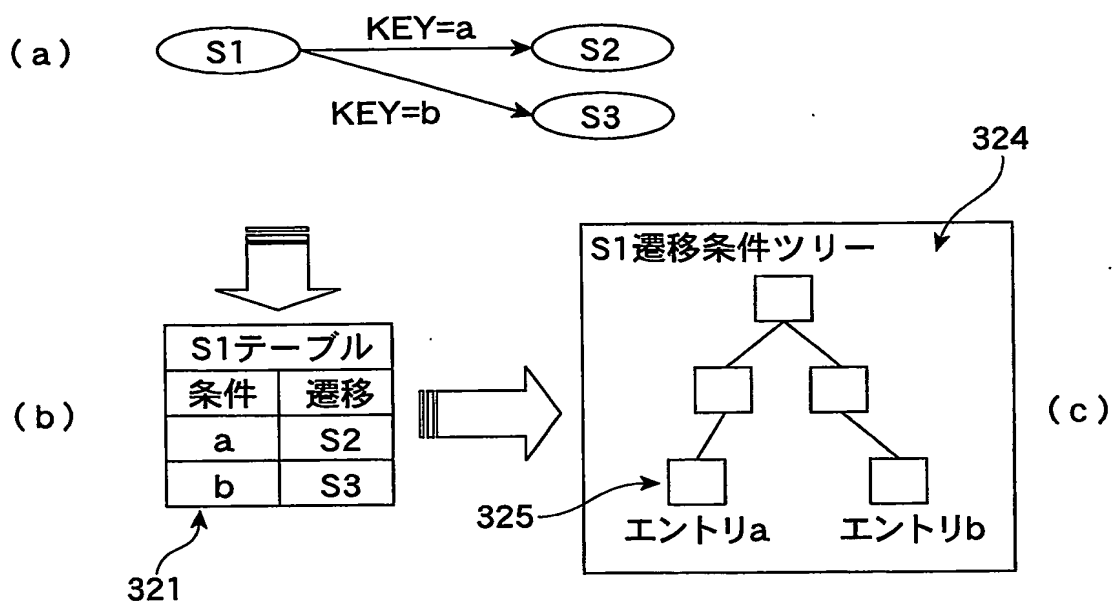
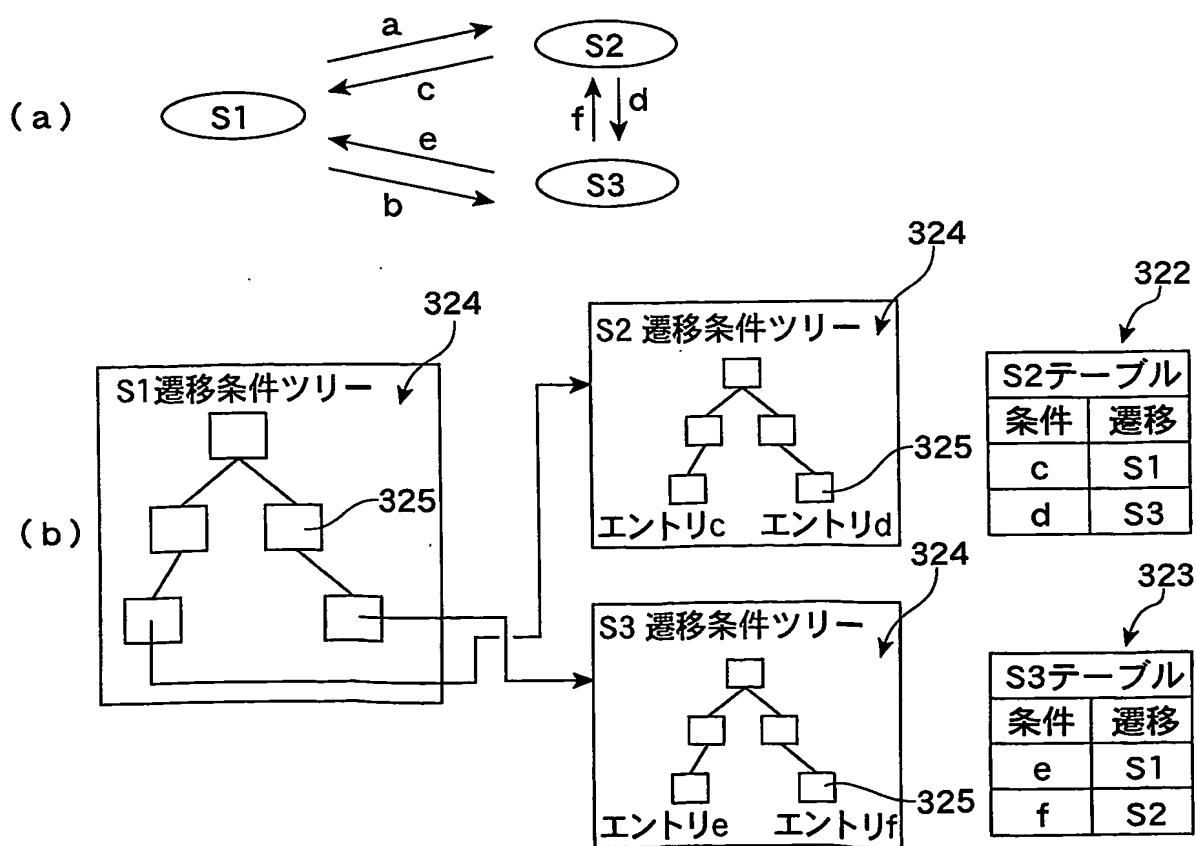
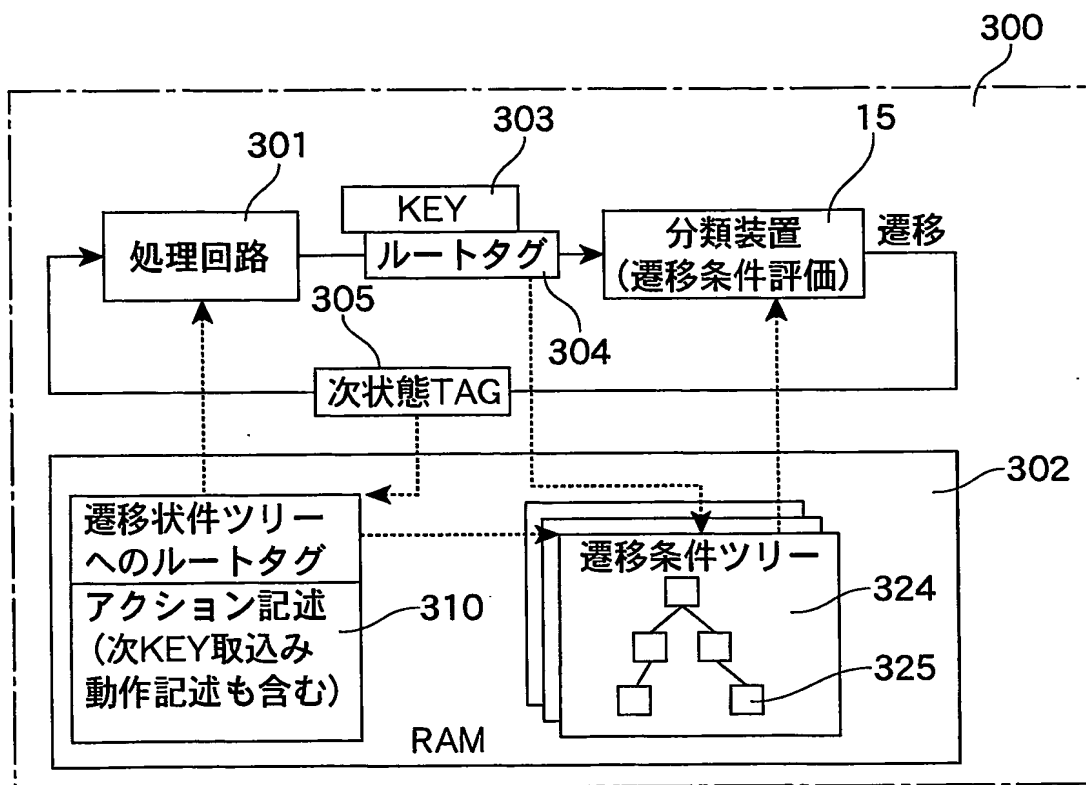


図 7 3



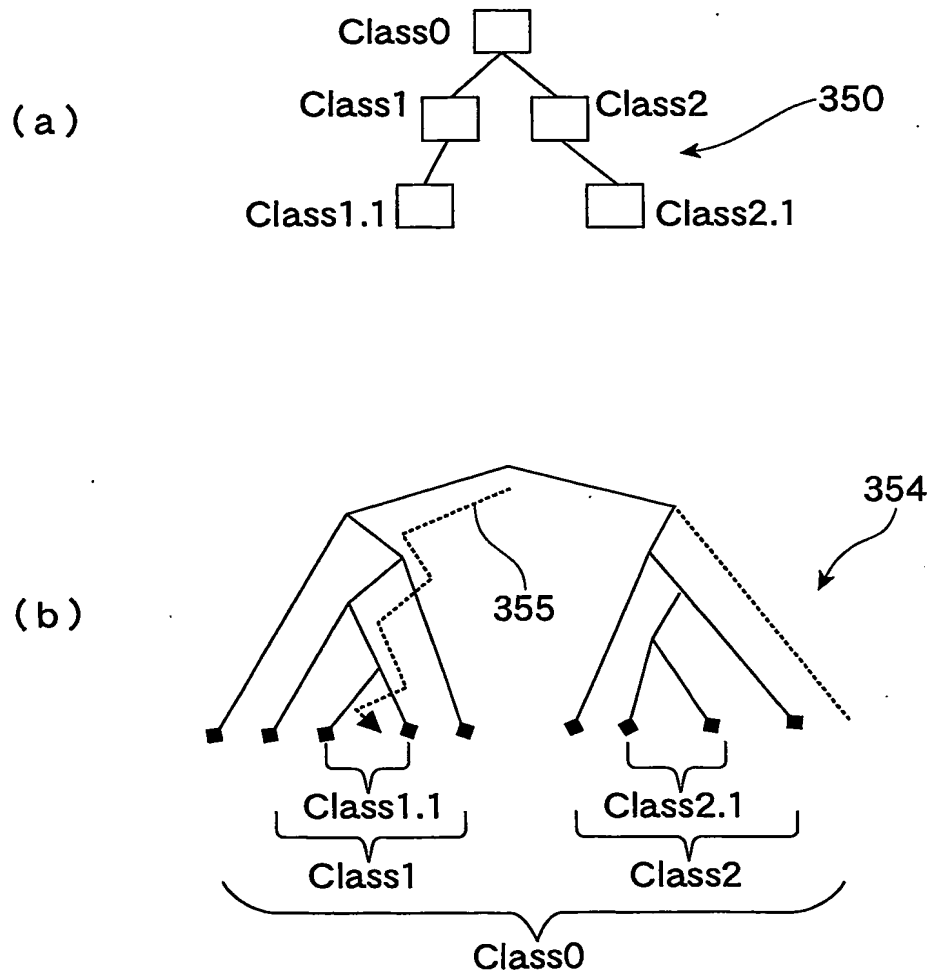
50 / 53

図 7 4



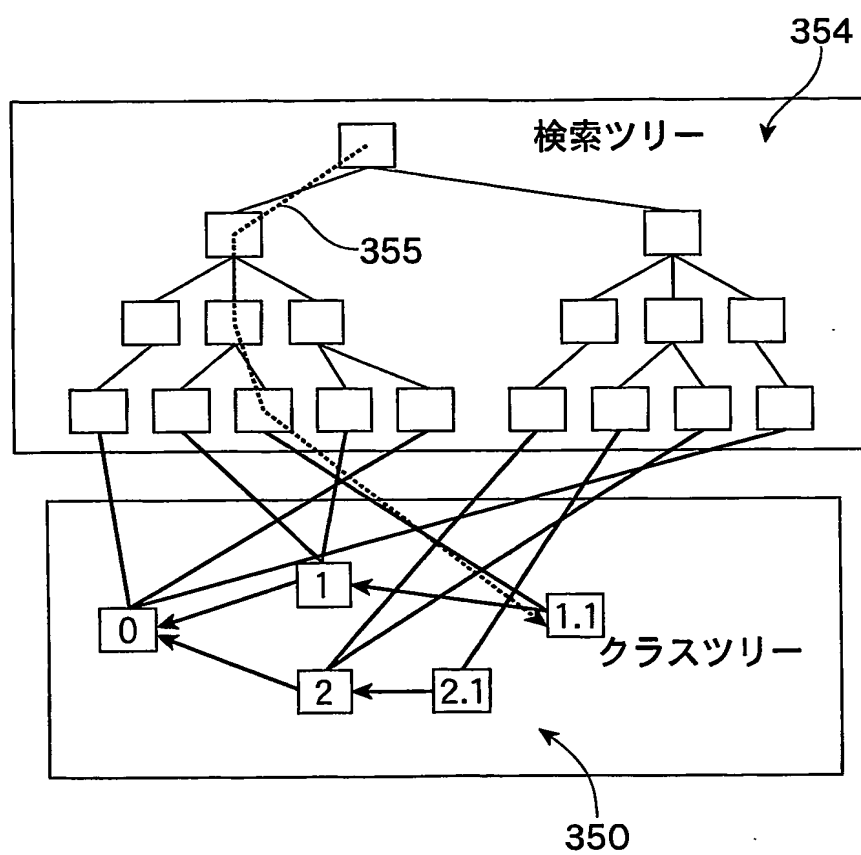
5 1 / 5 3

図 7 5



5 2 / 5 3

図 7 6



5 3 / 5 3

図 7 7

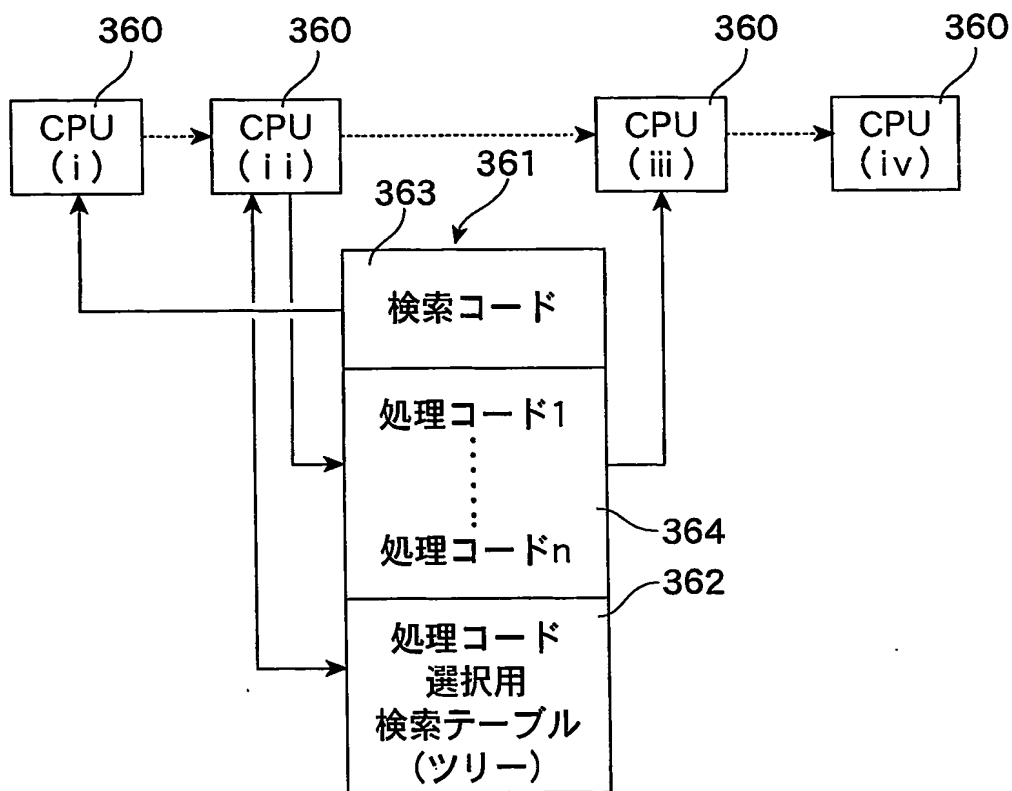
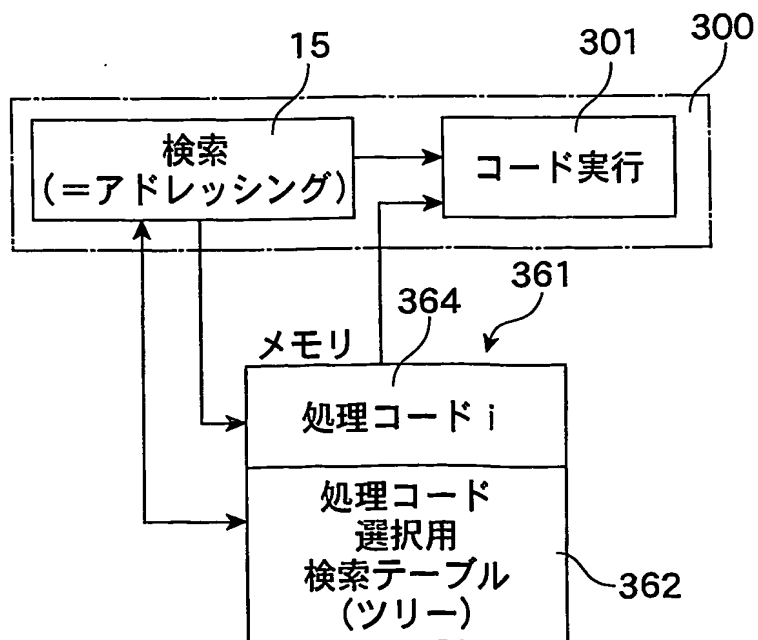


図 7 8



INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/12700

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl⁷ H04L12/56, G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl⁷ H04L12/56, G06F17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1926-1996 Toroku Jitsuyo Shinan Koho 1994-2003
Kokai Jitsuyo Shinan Koho 1971-2003 Jitsuyo Shinan Toroku Koho 1996-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 10-257066 A (Mitsubishi Electric Corp.), 25 September, 1998 (25.09.98), Abstract; Claim 1; Par. Nos. [0020] to [0032]; Figs. 1 to 3 (Family: none)	1, 3, 9, 10, 14, 19-21, 23, 29-33, 39, 42, 43, 45-47, 49-51 2, 22
Y A		4-8, 11-13, 15-18, 24-28, 34-38, 40, 41, 44, 48
Y	JP 2002-16638 A (Mitsubishi Electric Corp.), 18 January, 2002 (18.01.02), Figs. 2, 7 (Family: none)	2, 22

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
---	--

Date of the actual completion of the international search
20 November, 2003 (20.11.03)

Date of mailing of the international search report
09 December, 2003 (09.12.03)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/12700

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Kazuyuki KASHIMA, Keiichi SODA, Yoshihiro MORIKI, Tatsuki ICHIHASHI, Tetsuya ABE, 'IP Routing Table no. Hardware ni yoru Kosoku Kensaku Hoshiki', 1998 Nen The Institute of Electronics, Information and Communication Engineers Tsushin Society Taikai B-7-123, 07 September, 1998 (07.09.98)	1-51
A	JP 8-212790 A (Kawasaki Steel Corp.), 20 August, 1996 (20.08.96), Claim 1 (Family: none)	17, 41

A. 発明の属する分野の分類 (国際特許分類 (IPC))
Int. C1' H04L12/56, G06F17/30

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))
Int. C1' H04L12/56, G06F17/30

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1926-1996年
日本国公開実用新案公報 1971-2003年
日本国登録実用新案公報 1994-2003年
日本国実用新案登録公報 1996-2003年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X Y A	JP 10-257066 A (三菱電機株式会社) 1998.09.25 要約, 請求項1, 第0020段落から第0032段落, 第1図から第3図 (ファミリーなし)	1, 3, 9, 10, 14, 19-21, 23, 29- 33, 39, 42, 43, 45-47, 49-51 2, 22 4-8, 11-13, 15 -18, 24-28, 34 -38, 40, 41, 44, 48

☒ C欄の続きにも文献が列挙されている。

☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの
「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
「O」 口頭による開示、使用、展示等に言及する文献
「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献
「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」 同一パテントファミリー文献

国際調査を完了した日

20.11.03

国際調査報告の発送日

09.12.03

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)
郵便番号100-8915
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)
石井 研一



5X 3047

電話番号 03-3581-1101 内線 3596

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y	J P 2 0 0 2 - 1 6 6 3 8 A (三菱電機株式会社) 2 0 0 2 . 0 1 . 1 8 第 2 , 7 図 (ファミリーなし)	2, 22
A	鹿島 和幸, 曾田 圭一, 森木 嘉宏, 市橋 立機, 安部 哲也, 「IPルーチングテーブルのハードウェアによる高速検索方式」, 1998年電子情報通信学会 通信ソサイエティ大会 B-7-123, 1 9 9 8 . 0 9 . 0 7	1-51
A	J P 8 - 2 1 2 7 9 0 A (川崎製鉄株式会社) 1 9 9 6 . 0 8 . 2 0 請求項 1 (ファミリーなし)	17, 41

特許協力条約に基づく国際出願願書

原本（出願用） - 印刷日時 2003年10月02日（02.10.2003）木曜日 17時21分01秒

VIII-3-1	<p>先の出願の優先権を主張する国際出願日における出願人の資格に関する申立て</p> <p>出願人が優先権主張の基礎とされた先の出願の出願人と同一でない場合、又は先の出願の出願日以後に出願人の氏名又は名称が変更された場合において、以下の先の出願に基づく優先権を主張する国際出願日における出願人の資格に関する申立て（規則4.17(iii)及び51の2.1(a)(i)(ii)）</p>	<p>本国際出願に関し、</p> <p>以下の事実により、株式会社インフォーエスは先の出願特願2002-291708に基づく優先権を主張する資格を有している。</p>
VIII-3-1 (vii)		<p>2003年03月07日（07.03.2003）付で、吸収合併によってなされた有限会社ディシーエルから株式会社インフォーエスへの資格の移転</p>
VIII-3-1 (ix)	<p>本申立ては、次の指定国のためになされたものである。：</p>	<p>すべての指定国</p>